UNIVERSIDAD CARLOS III DE MADRID

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

TESIS DOCTORAL

DISEÑO Y EVALUACIÓN DE LA INTERCONEXIÓN DE REDES PEER-TO-PEER ESTRUCTURADAS USANDO UNA TOPOLOGÍA JERÁRQUICA

Autor: **Isaías Martínez Yelmo**
Ingeniero en Telecomunicación, Máster en Ingeniería Telemática

Directora: **Carmen Guerrero López**
Doctora en Ingeniería Informática

Leganés, Marzo de 2010

UNIVERSIDAD CARLOS III DE MADRID

DEPARTMENT OF TELEMATICS ENGINEERING

Ph.D. THESIS

DESIGN AND EVALUATION OF INTERCONNECTING STRUCTURED PEER-TO-PEER NETWORKS WITH A HIERARCHICAL TOPOLOGY

Author: **Isaías Martínez Yelmo, Ms.C.**

Supervisor: **Carmen Guerrero López, Ph.D.**

Leganés, March 2010

DISEÑO Y EVALUACIÓN DE LA INTERCONEXIÓN DE REDES PEER-TO-PEER ESTRUCTURADAS USANDO UNA TOPOLOGÍA JERÁRQUICA

DESIGN AND EVALUATION OF INTERCONNECTING STRUCTURED PEER-TO-PEER NETWORKS WITH A HIERARCHICAL TOPOLOGY

Autor: Isaías Martínez Yelmo

Directora: Carmen Guerrero López

Tribunal nombrado por el Mgfco. y Excmo. Sr. Rector de la Universidad Carlos III de Madrid, el día ___ de Marzo de 2010. Firma del tribunal calificador:

Firma:

Presidente:

Vocal:

Vocal:

Vocal:

Secretario:

Calificación:

Leganés, ___ de Marzo de 2010.

A mis padres y abuelos,
muchas gracias por todo.

Pocos hombres tienen la fuerza de carácter suficiente
para alegrarse del éxito de un amigo sin sentir cierta envidia.

– Esquilo (525 AC - 456 AC)

Las personas no son recordadas por el número de veces que fracasan,
sino por el número de veces que tienen éxito.

– Thomas Alva Edison (1847-1931)

Agradecimientos

Aunque mi prosa en castellano la tengo un poco olvidada, no puedo desaprovechar la ocasión para recordar a todas aquellas personas que han hecho posible que esta Tesis doctoral haya llegado a buen fin.

Como no puede ser de otra manera, el apoyo de mi familia ha sido inestimable en este largo viaje. Mis padres (Isaías y Nina) y mis abuelos (Dimas y Nieves) siempre me han apoyado y se han preocupado de mí durante todo este tiempo. Supongo que ahora todos ellos podrán decir que su hijo ya es doctor aunque no todos estén presentes para decírmelo personalmente. Además de ellos he tenido a toda mi familia dándome su apoyo ya sean tíos, primos o sobrinos. Mención especial a estos últimos que siempre me han alegrado y contagiado de su felicidad e inocencia en cualquier lugar y momento que he podido estar con ellos.

Tengo que darle las gracias a Carmen, mi directora de Tesis, por la libertad que ha dado a la hora de realizar este trabajo y por haber confiado en mí. Los comienzos no fueron fáciles pero creo con esfuerzo y trabajo hemos llevado a buen puerto este proyecto en el que tanta ilusión hemos derrochado.

Junto con Carmen ha habido otra mucha gente que me ha ayudado en este largo y costoso trabajo, los más importantes han sido Andreas Mauthe, Alex Bikfalvi y Roberto González Sánchez.

A toda la gente del Departamento de Ingeniería Telemática tengo que agradecerle el haberme soportado en el día a día y el haber compartido tantos y tan buenos momentos, incluso los malos cuando ha sido necesario. Tengo que mencionar especialmente a todos aquellos que me han enseñado como desenvolverme en este mundo de la investigación a parte de brindarme su amistad como Ignacio, David, Paco, María, Arturo, José Félix, Jaime, Albert o Marcelo. Especialmente, quiero recordar a la gente con los que durante muchos años no solo he compartido lugar de trabajo, también estudios, fiestas, comidas, cenas, viajes y muchos grandes momentos con sus anécdotas incluidas (como no podría ser de otra manera). Este elenco de personas son: Carlos, Richi, Pablo, Manolo, Iván, Isaac y Antonio. Tampoco puedo olvidarme de sus respectivas ya que aguantarnos no es tarea fácil: Tere, Rocío, Raquel, Mar y Julia. Además rondando por el Departamento hay mucha más gente

de la cual no puedo olvidarme: Elvira, Marco, Mika, Goyo, David, Sophie, Rubén y Ángel (más conocidos como el gemelo bueno y el gemelo malo, el orden es importante :D), Mark, Flora, Amparo y mucha más gente que seguro me dejo en el tintero.

Esta Tesis ha sido desarrollada en gran parte gracias a la Red de Excelecia Content financiada por la comisión Europea dentro del 6^o Programa Marco. En este proyecto he conocido a mucha gente la cual realmente me ha ayudado mucho en sacar este trabajo adelante. Son muchos por lo que no voy a enumerarlos, simplemente destacar que están en mi memoria.

Además no todo el mundo del que he recibo apoyo está relacionado con la universidad, también hay mucha gente sin la cual este camino hubiera sido imposible de soportar. El orden no importa ya que son muchos y ninguno está por encima de otro. En primer lugar el que equipo de madrileños (en su mayoría de Carabanchel) por España y parte del Mundo: David, Fernando, Herraiz, Casañas, Matías, Andrés, Félix y Culofino; gracias por tan grandes momentos juntos, no sé que hubiera hecho sin vosotros. Por supuesto, tampoco me puedo olvidar de mi grupo de turismo rural: Patxi, Cristina, Adrián, Sergio y todos aquellos que se nos han unido en nuestras escapadas. Hay gente a la que no veo tan frecuentemente como quisiera, pero que siempre las tengo presentes: Laura, Aroa, Rubén, mis Marías, Urbano, Marta, Mois, Patri, los Chikys, los Cokis, los Matrix, mis locas y una larga lista gente que no puedo enumerar por falta de tiempo más que por falta de ganas.

Resumiendo muchas gracias a todo el mundo por vuestro apoyo, amistad y comprensión que han hecho que este trabajo haya sido lo más llevadero y agradable posible.

Abstract

The traffic in the Internet is evolving continuously. In the 20^{th} century, the most traffic supported by Internet Service Providers (ISP's) was related with web traffic. However, nowadays, the traffic in the Internet has evolved drastically; now, most of the traffic in the Internet is Peer-to-Peer traffic. This fact changes completely the situation in comparison with the end of the previous century, thus the relevance of the Peer-to-Peer paradigm nowadays is evident.

The adoption of Peer-to-Peer overlay networks was firstly motivated for its usage in file-sharing applications but the applicability of Peer-to-Peer overlay networks is not only limited to this kind of applications. In fact, Peer-to-Peer overlay networks are suitable for the development of any distributed application or service since they allow the allocation and retrieval of information in a distributed fashion among a set of nodes. However, each overlay network has its own structure and mechanisms to distribute the information among all nodes. Additionally, each Peer-to-Peer overlay network implementation usually defines its own packet format. Therefore, the interoperability among different overlay networks is not possible.

This Thesis defines a mechanism to allow the exchange of information among different structured Peer-to Peer overlay networks, concretely DHT (Distributed Hash Table) overlay networks. This mechanism is based on both a common packet format, which assures the interoperability among different overlay networks, and on a hierarchical architecture. This hierarchical architecture has two levels of hierarchy. The lower level of the hierarchy is composed by the different overlay networks that want to be interconnected. Each one of these overlay networks can use any DHT overlay network with no restrictions. In addition, each overlay network in the lower level has at least one special peer, called super-peer. These super-peers are attached to the top level. This top level is named Interconnection Overlay and it is composed by just one overlay network. The purpose of super-peers is to route the queries among different overlay networks and they use the Interconnection Overlay to achieve this objective. In this Interconnection Overlay, the location information of each one of the super-peers and the overlay network represented by them are stored. Therefore, super-peers can forward the queries with the information stored in the Interconnection Overlay. If a resource placed in other overlay network wants to be obtained, a peer has to forward the query to

its super-peer. The super-peer gets from the Interconnection Overlay the information about the super-peer that takes care of the destination overlay network and forwards this request. Finally, the super-peer in the destination overlay network looks for the desired resource and once is retrieved the answer is sent back to the requester.

The proposed architecture is mathematically analysed to obtain is performance in term of hops and number of overlay routing entries in peers. Furthermore, the proposal is validated with a simulation tool to assure that the assumptions in the analytical model have been enough accurate. Finally, a real implementation over a controlled environment demonstrates the applicability and viability of the proposal and allows removing many of the original assumptions. The scenarios for the simulation analysis and the evaluation of the implementation have been designed carefully in order to define conditions as similar as possible to the real world.

Keywords: *Peer-to-Peer, overlay networks, hierarchical Peer-to-Peer, hierarchical overlay networks, interconnection of Peer-to-Peer networks, P2PSIP*.

Resumen

El tráfico en Internet está evolucionando continuamente. En el siglo XX, la mayor parte del tráfico en Internet soportado por los Proveedores de Servicios de Internet (ISP's en terminología anglosajona) estaba relacionado con el tráfico web. Sin embargo, actualmente, el tráfico en Internet ha evolucionado drásticamente. La mayor parte del tráfico en Internet es tráfico Peer-to-Peer. Este cambio cambia completamente la situación en comparación con el siglo anterior, de ahí la relevancia del paradigma de las redes Peer-to-Peer.

La adopción de las redes Peer-to-Peer esta principalmente motivada por su uso en aplicaciones de compartición de ficheros pero la aplicación de las redes Peer-to-Peer no está solo limitada al paradigma de compartición de ficheros. De hecho, las redes Peer-to-Peer son adecuadas para el desarrollo de cualquier servicio o aplicación distribuida ya que permiten almacenar información de manera distribuida entre un conjunto de nodos. Además, también permiten recuperar esa información cuando sea necesario. Una aplicación relevante basada en redes Peer-to-Peer es Skype la cual permite un servicio de VoIP entre varios millones de personas. Sin embargo, la interoperabilidad entre diferentes redes Peer-to-Peer no ha sido resuelta todavía. Cada red Peer-to-Peer define su propio mecanismo y su propio formato de paquete. Por lo tanto, sería deseable definir algún mecanismo que permita el intercambio de información entre diferentes redes Peer-to-Peer.

Esta Tesis define un mecanismo que permite el intercambio de información entre diferentes redes Peer-to-Peer estructuradas, concretamente redes overlay basadas en DHT's (Distributed Hash Tables). Este mecanismo está basado en un formato común de paquete, que asegura la interoperabilidad entre diferentes redes overlay, y en una arquitectura jerárquica. Esta arquitectura jerárquica está compuesta por dos niveles de jerarquía. El nivel más bajo de la jerarquía está compuesto por las diferentes redes overlay que desean estar interconectadas. Cada una de estas redes overlay puede usar cualquier DHT, no existe ninguna restricción al respecto. Al menos un super-peer existe en cada una de estas redes overlay del nivel inferior; además, estos super-peer también participan en el nivel superior. Al nivel superior se le conoce como Red de Interconexión y está compuesto sólo por una red overlay. Su función es similar al servicio de DNS pero en el área de las redes Peer-to-Peer. En la Red de Interconexión se guarda la información de localización de cada uno de los super-peers y también el dominio o la overlay a la que representan. Por lo tanto, si un recurso de otra red quiere

ser recuperado, un peer tiene que reenviar la petición a su super-peer. El super-peer consulta a la Red de Interconexión para localizar al super-peer que se hace cargo de la red destino donde se encuentra el recurso deseado y una vez que se localizar al super-peer, se le reenvía la petición. Finalmente, el super-peer en la red destino buscar el recurso deseado y lo envía de vuelta al peer que origino la petición.

La arquitectura propuesta ha sido estudiada analíticamente para asegurar que el rendimiento es razonable en comparación con otras redes Peer-to-Peer. Además, la propuesta es validada con una herramienta de simulación para asegurar que las asunciones en el modelo analítico no afectan en un escenario más general. Finalmente, una implementación real sobre un entorno controlado es mostrada para demostrar la aplicabilidad y viabilidad de la propuesta. Los escenarios para las simulaciones y la verificación de la implementación han sido diseñados con especial cuidado para tener unas condiciones lo más cercanas posibles a escenarios reales.

Palabras clave: *Peer-to-Peer, redes overlay, Peer-to-Peer jerárquico, redes overlay jerárquicas, interconexión de redes Peer-to-Peer, P2PSIP*.

Contents

List of Figures

List of Tables

Vivir no es sólo existir,
sino existir y crear,
saber gozar y sufrir
y no dormir sin soñar.
Descansar, es empezar a morir.

Gregorio Marañon (1887-1960)

Chapter 1

Introduction

The Internet traffic patterns have changed in the last years. In the past century, web traffic was the most common traffic in ISP's and backbones; however, nowadays applications and services based on Peer-to-Peer overlay networks are the most common traffic in the Internet. This change in the traffic patterns is especially due to the appearance of file-sharing applications that make use of these Peer-to-Peer overlay networks, but other applications such as Skype also contribute to the current Peer-to-Peer traffic.

Many types of Peer-to-Peer overlay networks have been defined along the last years and also many applications have made use of them. Each one of them has its advantages and disadvantages and depending on the usage scenario one solution is more suitable than other ones. Nevertheless, different Peer-to-Peer overlay networks cannot exchange information among them since they do not share the same topological structure, routing mechanisms and payloads. This problem of incompatibility occurs not only among different overlay networks, it also happens among implementations of the same kind of Peer-to-Peer networks since the payload or configuration parameters are setup in a different way. Thus, it would be desirable to have some mechanism that would allow the exchange of information among different Peer-to-Peer overlay networks.

This Thesis makes a proposal of a hierarchical Peer-to-Peer network that allows the exchange of information among different kinds of overlay networks in order to address some of the deficiencies previously commented. The main objectives of this Thesis are as follows:

- *Exchange of information among different overlay networks:* the exchange of information among different overlay networks must be possible in any case.
  - *The overlay networks can have different topologies:* not only must be possible the exchange of information among overlay networks with the same topology, it must be also necessary to allow the exchange of information among overlay networks with different kinds of topology. Therefore, some mechanism must be defined to allow this exchange.

- *Use of standards:* the exchange of information amount peers of different overlay networks should use some predefined mechanism for an ordered exchange of information. The best way to assure the adoption of a specific mechanism is to use a standard that can be used as reference.
- *Validation of the proposed solution:* the proposed solution must be validated in order to assure that all the design process has been correctly achieved and its results fulfil the desired expectations. Two different mechanisms can be used:
  - *Simulation:* the usage of simulators can give a first overview of the real performance but with some assumptions about the under layers.
  - *Implementation:* a real implementation that behaves as expected validates the proposal and demonstrates the viability of the solution over a real TCP/IP stack.

The content of this Thesis is structured on several parts. Part I gives an overview of the current state of the art related with Peer-to-Peer overlay networks. This part is composed by several chapters. Chapter 2 gives an overview of the different kinds of Peer-to-Peer applications. Several classifications are possible and some of them are reviewed. Finally, the chapter focuses on the Peer-to-Peer overlay networks that are more relevant to this Thesis; they are the structured overlay networks. An overview of some of them is provided as well as the main problem that Peer-to-Peer overlay networks have to support, the churn rate. Chapter 3 focuses on the paradigm of hierarchical overlay networks. The first Peer-to-Peer networks are known as flat overlay networks since all peers have the same role and they present a flat architecture. Although, it is demonstrated that the Peer-to-Peer technologies have good properties, it is always desirable to improve these properties and especially their scalability. With this way of thinking, the hierarchical Peer-to-Peer overlay networks were born. Different kinds of hierarchical overlay networks are explored, each one designed with a different objective and an overview of advantages and disadvantages of these topologies is provided. Especial attention is provided to the hierarchical overlay networks based on super-peers, since our proposal would make use of this role. Finally, chapter 4 introduces P2PSIP. P2PSIP is an IETF Working group that it is currently defining a standard protocol to support almost any Peer-to-Peer overlay network and it is especially focused on DHT overlay networks. It is very important to mention this work since it provides an excellent framework to define an interoperability among different Peer-to-Peer overlay networks.

Part II contains the design of the solution and an estimation of its performance based on a mathematical analysis. First of all, in chapter 5, a summary of the goals and design considerations for this Thesis is provided. Chapter 6 provides our proposal named H-P2PSIP which consists of a hierarchical DHT overlay network with two levels of hierarchy. The lower level is composed by the different overlay networks that want to establish communications with other overlay networks, whereas the top level of the hierarchy has only one overlay network named Interconnection Overlay. If we want to understand this architecture, it is important to highlight that it makes use of super-peers. At least one super-peer must exist per overlay network in the lower level and these super-peers form the Interconnection Overlay in the upper lever. The information stored in the Interconnection Overlay is very specific, the location information of each super-peer and the overlay network or domain that each one is repre-

senting. Therefore, the Interconnection Overlay can be used to get the necessary information to route the queries to the destination overlay network. If a peer participating in an overlay network in the lower level wants to retrieve information from other overlay network, it must forward the query to its super-peer. This super-peer can find the super-peer taking care of the destination overlay network, since this information is stored in the Interconnection Overlay, and the query can be forwarded to the discovered super-peer. Finally, the super-peer in the destination overlay network localizes the resource and sends it back to the requester. The idea is quite simple however is necessary to define some elements:

- *Data structures:* to assure the localization and correct storage of the information in the hierarchical architecture.
- *Signalling:* that allows an ordered way of exchanging the information.
- *Common payload:* to permit the exchange of information among peers of different overlay networks.

More details can be found in the previously mentioned chapters or in these references: [MYBG+08a], [MYBC+09]. Chapter 7 provides an analytical model of our proposal. It is a simple analysis that assures the scalability of the solution and shows where are the possible bottlenecks in the proposal. The publications related with this analysis are [MYCGM08], [MYGCM09].

Part III provides a validation to the design realized in Part II. In chapter 8, the proposal is validated using a Peer-to-Peer simulation framework which has been modified to support our hierarchical architecture. The results show that the architecture works as expected. The publications related with this validation are [MYBG+08b], [MYBG+08a], [MYBC+09]. Finally, the last stage of our research, once the previous steps have been concluded, it is to do a real implementation of our proposal. Chapter 9 gives an overview of our implementation and the results associated with this work. This implementation gives a proof of our proposal since the performed experiments are based on a real TCP/IP stack and a population of hundred of peers.

The results obtained in Parts II and III are compared with a flat counterpart. This mechanism is adopted since the information provided for flat overlay networks is much more complete in terms of deployed mathematical analysis, available simulation tools and available implementations. Furthermore, the actual flat DHT overlay networks can be extended to our proposal, so it is interesting to compare both of them.

Finally, Part IV finishes this Thesis. Chapter 10 presents the resulting conclusions of the contributions of this Thesis, whereas chapter 11 gives some relevant future work topics that could be interesting to explore in the near future.

Other publication related with Peer-to-Peer networks but focused in other topics are [MYCGM07], [CGNMY07] and [CGMYN07].

Part I

State of the Art

Todo hombre tiene tres variedades
de carácter: el que realmente tiene,
el que aparenta, y el que cree tener.

Jean Baptiste Alphonse Karr
(1808-1890)

Chapter 2

Peer-to-Peer Overlay Networks

This chapter provides a description of the state of the art on Peer-to-Peer overlay networks, describing current designs and solutions as well as identifying open issues. This chapter goes from the most general overview of Peer-to-Peer overlay networks to the most specific details that would be later necessary for the design of our proposal.

2.1 Introduction

Peer-to-Peer (p2p) networks are a paradigm that appeared in this decade and they are in continuous evolution. The concept of *overlay network* is not novel; it can be defined as a computer network built on top of another network (such as IPv6 over IPv4 or IP multicast). Peer-to-Peer networks are a particular case of overlay networks where all participants in the overlay network (named *peers*) contribute with their own resources to fulfil an objective in a distributed fashion. The most common application related with Peer-to-Peer networks is file-sharing (Emule[1], Bittorrent[2], etc). Other applications are also interesting and their popularity is increasing continuously like Skype[3] [BS06] (probably the best known VoIP application nowadays). A classification of the different applications that can be supported by Peer-to-Peer networks can be found in [MKL+02] or [ATS04]. Each overlay network has its advantages and disadvantages; thus, depending on the application and the usage scenario, one overlay network is more suitable than other ones.

This chapter gives an overview of the different types of overlay networks depending on several aspects and focuses on the more relevant work related with this Thesis.

[1] http://www.emule-project.net
[2] http://www.bittorrent.com
[3] http://www.skype.com

2.2 Classification of Peer-to-Peer Overlay Networks

There is not a clear classification about Peer-to-Peer Overlay Networks. This classification can be done according to the infrastructure of the Peer-to-Peer overlay network, the way used to store and retrieve the information or depending on how the information is structured on the overlay network. Other classifications may exist but these ones are the most widely accepted.

2.2.1 Classification based on the role of peers

This classification is based on the infrastructure that must be deployed to support the store and retrieval functionalities in a Peer-to-Peer overlay network. Although a common classification does not exist, common points can be found in the literature [BWDD02], [ATS04].

2.2.1.1 Centralised Peer-to-Peer networks

Centralised Peer-to-Peer overlay networks have a central repository (composed by one or several special entities) storing all the resources maintained by all the peers participating in the overlay network. Therefore, the resource discovery for peers consists in querying the central repository to get the location of the desired resources. Once the location is obtained, the peers contact one another and share the information in a distributed fashion. The best examples of this kind of Peer-to-Peer overlay networks are Napster[4] and eMule[5] [KBPK05].

2.2.1.2 Decentralised Peer-to-Peer networks

All peers in decentralised Peer-to-Peer networks play the same role in the overlay network. They perform the same tasks, acting both as clients and as servers. The resource discovery is performed in a distributed fashion as well as sharing the information, central servers are not necessary for any task. Therefore, they offer a tool to build fully distributed applications and services. The most representative overlay networks of this type are Gnutella 0.4 [6] [gnu00] and the KAD network [BB06] (decentralized version of eMule), both mainly used for file-sharing.

2.2.1.3 Hybrid Peer-to-Peer networks

In Hybrid Peer-to-Peer networks, not all peers assume the same role and responsibilities. A subset of them can have special functions and tasks, these special peers are usually named ultra-peers or super-peers. Furthermore, some central entities can exist to support

[4] http://en.wikipedia.org/wiki/Napster
[5] http://www.emule-project.net
[6] http://en.wikipedia.org/wiki/Gnutella

some special features. These special features are usually related with the enrolment of peers since it is the weakest point in the security of Peer-to-Peer networks [Wal03], [SM02]. Any Peer-to-Peer system based on super-peers belongs to this group. Probably, the best known application and protocol is Gnutella 0.6 [gnu02].

2.2.2 Classification based on the structure of overlay networks

A second way of classifying Peer-to-Peer overlay networks is according to the structure maintained in the overlay network to store and recover information. In the next paragraphs, a short introduction to this classification is given.

2.2.2.1 Unstructured Overlay Networks

Unstructured overlay networks are those ones that have peers, which do not maintain any structure among them in order to establish any communication or exchange of information. The way to discover nodes is usually based on a bootstrap server and also on getting the information from neighbours discovered lately through the bootstrap server. Because no structure is maintained, the way of performing the queries is based on flooding, but with a Time to Live (TTL) to avoid traffic storms, or random walks [LCC+02], [GMS04]. This kind of overlay networks can be completely distributed like Gnutella 0.4 [gnu00] or hybrid like Gnutella 0.6. Basically, the new version of Gnutella includes the concept of super-peers to manage the location of resources in the overlay network. Super-peers are especially useful in this kind of networks because the number of nodes taking care about the resources is smaller, therefore the flooding requests or random walks are smaller and the consumed traffic decreases whereas the scalability of these overlay networks increases.

The drawback of this kind of overlay networks is the dependency on reaching the nodes trough flooding or random walk algorithms. These mechanisms do not assure to reach all the peers in an overlay network; thus, some queries may not be answered if the peer with the desired resource is not reached. However, the great advantage of these overlay networks is the flexibility in performing the queries. Keyword searches or regular expressions can be applied over the name of the resources or their meta-data to perform queries. This is possible because any restriction is applied when the resources are stored or searched in an unstructured overlay network; this fact does not apply in structured overlay networks (section 2.2.2). The most well know applications running this kind of Peer-to-Peer overlays are Gnutella [gnu02], FreeNet[7] and Limware[8].

2.2.2.2 Structured Overlay Networks

Structured Overlay Networks have the property of supporting a structured topology. This fact means the placement and storage of the information in the overlay network is not ran-

[7] http://freenetproject.org/
[8] http://www.limewire.com/

dom like in unstructured overlays. These kinds of networks are usually based on Distributed Hash Tables (DHT's). Each peer in the DHT overlay network is identified by an Identifier (ID). Furthermore, the resources stored in this kind of networks are also identified with a similar ID named Resource-ID; this Resource-ID is used to store the resource in the overlay network. The way to store the resources is based on placing the resources in the node with the ID closest to the Resource-ID. Therefore, in order to recover the resources from the overlay network, it is necessary to build overlay routing tables that allow reaching any peer in the overlay network. Given a Resource-ID, the routing algorithm must route the queries to the peers with the ID closest to the Resource-ID until a peer is reached with the desired information. The differences among the different DHT overlay networks consist in their structure, how the routing tables are built and how the queries are routed to the destination. There are many DHT overlay networks: Chord [SMLN+03], Kademlia [MM02], CAN [RFH+01], Pastry [RD01], Tapestry [ZHS+04] or Bamboo [RGRK04]. In next section, some of them are studied deeply, since some of them are necessary to understand the work done in this Thesis and other ones are just interesting to compare the benefits and drawbacks of different strategies to build DHT overlay networks.

2.3 DHT based Overlay Networks

DHT's are an important part of this Thesis, so it is necessary to study them with a higher grade of details in order to understand better the contribution to the field of Peer-to-Peer networks.

Distribute Hash Tables are Peer-to-Peer overlay networks where peers and resources of the overlay network share the same flat key space. This means that a key (also called ID) identifies each peer in the overlay network as well as a Resource-ID identifies any resource. The resources are usually placed in the peer with the ID closest to the Resource-ID of a resource. However, other peers can store the resources although they are not the closest peers; it is said that they store replicas of the content. These peers are usually neighbours to the designated peer to store the resource since if a problem exists (churn, connectivity, etc), they can also provide the desired information because the routing algorithms route the queries in each hop to a closer peer to the resource.

One question that arises around this kind of overlay networks is how to assign the ID to the peers and resources. The peers need to have different ID, they must be uniquely identified. This is necessary since overlay routing tables are built to route queries; if two peers have the same ID, inconsistencies in the routing algorithm or in the routing tables can happen which would derive in undesired infinite loops when a query is routed to the destination. On the other hand, a Resource-ID can be shared among different resources, since the name of resources or associated meta-data information can be attached to the response of a query to differentiate the resources under the same ID. Considering the previous arguments, different strategies can be used for the generation of peer ID's. The first one is in a distributed way where each peer generates its own ID, which is a key of m bits. This key is usually based on the hash function of an attribute or capability of the peer (such as the IP address). However, the hash functions are not two-way functions so a hash function can

Taxonomy	Chord	Kademlia	Pastry	Can
Architecture	Uni-directional and circular Node-ID space	XOR metric for distance between points in the key space	Plaxton-style global mesh network	Multi-dimensional coordinate ID space
Lookup	Matching the Key-ID with the Node-ID	Matching Key-ID and Node-ID based routing	Matching of Key-ID and prefix in Node-ID	Mapping of a point P in the coordinate space using uniform hashing
Parameters	N-number of peers	N-number of peers b-number ($B = 2^b$)		N-number of peers d-number of dimensions
Routing Performance	$O\left(log_2 N\right)$	$O\left(log_B N\right) + c$ c = small constant	$O\left(log_B N\right)$	$O\left(dN^{\frac{1}{d}}\right)$
Routing State	$log_2 N$	$Blog_B N + B$	$2Blog_B N$	$2d$
Peers join/leave	$(log_2 N)^2$	$log_B N + c$	$log_B N$	$2d$

Table 2.1: *Performance of DHT based overlay networks*

generate the same output for different inputs; thus, some mechanism to detect duplicates is necessary. The other option is to have a central entity that provides the peer ID's, thus it can be assured that the ID is unique. Furthermore, it has some advantages from the point of view of security [Wal03], [SM02]. In relation with the Resource-ID's, a key can be assigned to a resource without any mapping function, but that resource only can be retrieved if the key is known in advance. In general, it is necessary to provide some mechanism to generate the key for the Resource-ID from the resource itself or from some associated information such as a name or URI. These mapping functions are usually based on hash functions. In fact, several mechanisms to publish information in DHT's exist in the literature as data indexing in Structured/DHT overlay networks [GEFB+04].

Table 2.1 summarises the characteristics of different DHT overlay networks. The most important features are illustrated like the lookup algorithm, characteristic parameters, Routing Performance (efficiency of the overlay routing usually measured in number of hops), Routing State (information stored in peers to perform the overlay routing operations usually measured in number of routing entries) or number of operations per join/leave action. Furthermore, in the next subsections, an overview of them is provided.

2.3.1 Chord

Chord [SMLN+03] is a Distributed Hash table with a flat name space of *m* bits. As usual, the peer ID's are composed by a key of *m* bits as well as the Resource-ID's. The mapping of peer ID's and Resource-ID's is usually based on consistent hashing, although any more secure mechanism can be provided if necessary. The flat name space is ordered in a ring topology with modulo 2^m. Keys are assigned to the *successor* peer of those keys; this means that each key is stored in the closest peer clockwise from that key. The routing table in a Chord peer is built as follows. Each peer must maintain a routing table called finger table. The i^{th} entry of the finger table consists in the identity of first peer that succeeds the peer by at least 2^{i-1} bits of difference with respect its own ID. Therefore, if n is a peer the i^{th} finger of n is: $n.finger\left[i\right] = successor\left(n + 2^{i-1}\right)$, where $i \in [1, m]$. The Routing Performance is given by the structure of the Peer-to-Peer overlay network and also by the mechanism used to populate and refresh the routing table of the peers; this Routing Performance is $O(log_2 N)$.

If a new peer joins in the overlay network, the new peer has to take care of the keys between its ID and its predecessor ID. The update of the routing tables is done periodically with a stabilization algorithm.

2.3.2 Kademlia

The Kademlia overlay network [MM02] is a Distributed Hash Table with tree-based routing that uses on a XOR metric to measure the distance among keys in the flat space domain. The XOR metric between two key is defined as their bitwise exclusive OR, this means $d(a,b) = a \oplus b = d(b,a)$. XOR is a unidirectional metric, this fact makes all lookups for the same key converge along the same path. Kademlia has a configuration parameter $B = 2^b$ that gives the number of bits that a query covers per hop. In order to achieve this objective, the routing table is divided in buckets where each bucket is associated to subset of the flat space as follows: $\left[j \cdot 2^{i \cdot b}, (j+1) \cdot 2^{i \cdot b}\right)$ where m is the number of bits of the flat space, $j \in (0, B]$ and $i \in [0, m/b)$ with m, j and i being natural numbers. Each bucket has a depth k that is the number of entries that can be stored in the routing table associated to each bucket interval. Therefore, the Routing Performance of Kademlia is $O(log_B N)$ and the Routing State needed is $O(B \cdot log_B N + c)$. The Kademlia is operated in iterative mode, this allows parallelizing the number of queried peers per hop and it also helps to populate the buckets with the information of the peers that answers to the queries in each hop. Thus, it is not necessary any stabilization algorithm to fix the routing table.

2.3.3 Pastry

Pastry [RD01] is a DHT that makes use of a Plaxton-like prefix routing. The flat name space is organized in a ring topology. Pastry has a configuration parameter $B = 2^b$. It is used as follows, each key in the DHT is considered as a sequence of digits with base B. The state stored in each peer is composed by a leaf set, a routing table and a neighbourhood set. The leaf set is the set of nodes with the $L/2$ numerically closest larger ID's and the L/2 closest smaller ID's (typical values are 2^b or 2^{b+1}). The routing table is formed by $log_B N$ rows where each one holds $B - 1$ entries. A routing entry in row n is referred to a peer whose ID shares own peer's ID in the first n digits, but whose $(n+1)^{th}$ digit is different from the own peer, and each possibility is stored in each of the $B - 1$ entries per row. Furthermore, a neighbourhood set is stored with the M nodes that are closest (according a proximity metric) to the own peer. The neighbourhood set is not normally used in routing messages; it is useful to populate the routing table with peers that are close according to the used proximity metric. When a query is being routed, the leaf set is checked. If any destination is found in the routing table, it is used to get the next hop. Considering how the routing table is built, Pastry has a Routing Performance of $O(log_B N)$ and the Routing State is $O(B \cdot log_B N)$.

2.3.4 Content Addressable Network (CAN)

CAN [RFH+01] is a Distributed Hash Table with a flat name space of *m* bits organized in a virtual d-dimensional cartesian coordinate space. Again, consistent mapping is used to map peers and resources to ID's if it is not desired additional security constraints. The virtual cartesian coordinate space is partitioned among all the peers in the overlay network and each node takes care of its own subspace. The routing table of each peer is composed by all the neighbours in the virtual d-dimensional space, the size of this routing table in each node is $2\dot{d}$, where d is the number of dimensions in the virtual cartesian coordinate space. The ideal case occurs when the coordinate space is divide in n equally zones where the Routing Performance is $d/4 \cdot N^{(1/d)}$; the Routing Performance can be approximated as $O(d \cdot N^{1/d})$. One characteristic is the fact that the size of the routing tables is independent with the number of peers in the overlay. Each time a peer joins in the overlay needs to take care of its portion in the coordinate space. This portion is obtained by splitting previous peer's zone in a half, the old peer takes care of a half and the new peer takes care of the other half. Furthermore, the new peer has to store all the keys associated to its portion of the coordinate space.

2.4 The churn rate in Peer-to-Peer Overlay Networks

The churn rate is an important factor in Peer-to-Peer overlay networks. In fact, if we consider only the previous section we could conclude that the overlay networks has an excellent scalability because most of them build their routing tables with a logarithmic complexity and the Routing Performance also depends with the logarithm of the number of participants in the overlay network. However, in real implementations, these results are not as good as expected. This fact is produced because in the previous analysis the churn rate has not been considered. The continuous number of arrivals and departures in the overlay networks makes the routing tables of the peers inconsistent and they do not reflect the real topology of the overlay network. This happens especially if the departures of peers have not been rightly signalled (produced by failures in the peers or by connectivity problems). An interesting study of churn in DHT overlay networks can be found [LSM+05]. With the help of a DHT overlay simulator, the behaviour of different DHT overlay networks under churn is studied. However, the churn used in this paper is modelled by exponential distributions, which is not true in real applications running in the Internet (see next paragraphs). Therefore, it would be interesting to have statistics about the behaviour of Peer-to-Peer overlay networks in real environments.

2.4.1 Churn in file-sharing applications

Peer-to-Peer networks have been usually deployed in file-sharing applications. Therefore, the information of real implementations of Peer-to-Peer networks is usually related with the topic of file-sharing. The most deployed overlay networks have been Gnutella [gnu02],

Bittorrent [9] and KAD [BB06] (the distributed version of eMule). There is a really interesting study about churn in [SR06] for all these overlay networks. In this study is show how the session time of peers is quite closed to log-normal and Weibull distributions. This means that there are many peers with short session times and some of them have long session times. However, the number of peers with long session times is not large and it cannot be considered as a heavy tailed distribution. Furthermore, it is also studied the peer uptime. The results show that is easy to have peers with a large uptime; however, a very important set of peers is unstable and it affects negatively to the overall performance of the overlay networks. Additional details can be found about Bittorrent in [PGES05], [EIP] and also about KAD in [SENB07c], [SBEN07], [SENB07a], [SENB07b]. In the last set of papers about KAD is measured how the arrival and departure interarrival time fit with a negative binomial distribution (not an exponential) and how the session time fits (again) with a Weibull distribution.

2.4.2 Churn in VoIP applications

It must be considered that there are more applications different from file-sharing applications that make use of Peer-to-Peer applications. In fact, one of the other more successful Peer-to-Peer applications is Skype[10], a VoIP application. In fact, the evolution of multimedia systems for streaming, Voice on Demand (VoD), etc are based on Peer-to-Peer networks because they reduce the bottlenecks in servers that are serving the media. Therefore, it is also interesting to study what happens with this kind of applications.

Skype is one of the most famous of VoIP applications in the Internet. The key point of Skype is its philosophy of install and use in almost any scenario or environment, this happens since it implements advanced NAT Traversal mechanisms that allows using the program in almost any network environment with independence of its security and restrictions. Skype [BS06] is based on an unstructured overlay network based on super-peers. These super-peers perform additional tasks; one of the most important is to play the role of relays if they have public addresses in order to provide NAT Traversal capabilities. Skype presents churn, however super-peers are specially selected if they fulfil some requirements, this fact gives a low churn for super-peer peers [GDJ06b]. Furthermore, it has been discovered how these super-peers follow day-time patterns similar to the patterns of their users. For instance, an employee in an enterprise switches his PC when arrives to the work and Skype runs in super-peer mode because fulfils the requirements for it. These requirements are usually long session times, public IP addresses, etc.

2.4.3 Mechanisms to handle the effects of churn

Basically, there are two basic ways to handle the churn, these ways are different on the strategy to update the overlay routing tables in order to remove the peers that are not

[9] http://www.bittorrent.org/beps/bep_0000.html
[10] http://www.skype.com

available at that moment. These two ways are Reactive and Periodic recovery, a detailed discussion about these strategies can be found in [RGRK04].

The Reactive recovery consists on update any change in the routing table of a peer if it is detected some invalid entry and this information is transmitted to its neighbours as soon as possible. On the other hand, the Periodic recovery updates the information periodically and the changes in the routing table are only sent to the neighbours if a period of time expires. It is demonstrated that reactive recovery performs better if the churn rate is small; however, if the churn rate is high, it does not perform so well since positive feedbacks cycles are created. Furthermore, when the churn rate is very high, the bandwidth consumption of the reactive mechanism also increases very much, which is not desirable. The periodic update performs well with high churn because unnecessary traffic and continuous changes in the routing tables are avoided. Furthermore, these changes would not be completely updated in any case because of the high churn. Therefore, depending on the expected churn of the overlay network, one strategy would be more suitable than other would.

2.5 Conclusions

In this chapter, an overview about the paradigm of Peer-to-Peer overlay networks is given. The Peer-to-Peer overlay networks allow the deployment of distributed services; however, there are many proposals and it is not easy to find the most suitable architecture to solve a problem. The different proposals can be classified as centralised, decentralised and hybrid Peer-to-Peer networks. The last ones are probably the most suitable to find a good trade-off between scalability, efficiency and security. Furthermore, with independence of the architecture of the Peer-to-Peer overlay network, they can be structured or unstructured. In this case, a trade-off arises among the resources used to precisely allocate the resources in a Peer-to-Peer overlay network and the cost of finding resources in an unknown topology. Finally, we take care on DHT overlay networks (structured overlay networks) seeing different proposals and their main advantages and drawbacks. Finally, we focus on the churn rate. The churn rate limits the efficiency and the scalability of Peer-to-Peer solutions and some mechanisms must be provided to avoid its impact, especially in structured overlay networks. The content of this chapter is necessary to follow the rest of the chapters in this State of Art Part and to understand the different elections made in our proposal.

Siempre que enseñes, enseña a la vez a dudar de lo que enseñes.

José Ortega y Gasset (1883-1955)

Chapter 3

Hierarchical Peer-to-Peer Overlay Networks

This chapter contains an overview of hierarchical Peer-to-Peer networks. This kind of overlay networks tries to improve some of the characteristics of the traditional flat overlay networks such as delay, Routing Performance and Routing State. This study is important since our proposal is based on a hierarchical overlay network. Therefore, it is necessary to review previous studies related with hierarchical overlay networks, their advantages and disadvantages as well as the possible open issues.

3.1 Introduction

Although Peer-to-Peer overlay networks are a good solution for distributed systems, its implementation is not a trivial issue, especially if we consider the trade-offs that must be taken into account in dealing with churn. Furthermore, if we want global applications that scales to a very large number of users, despite the logarithmic scalability of Peer-to-Peer overlays, perhaps it would be not enough in some cases. For instance, an overlay network has a minimum determined delay (bounded by the underlying network infrastructure), which is insufficient to deploy a certain application; however, some hierarchical overlay networks are specially designed to reduce this delay and they would be suitable to deploy the application instead of the flat counterpart. Therefore, it is necessary to explore new possibilities and hierarchical overlay networks are an interesting solution.

3.2 Motivation of Hierarchical Peer-to-Peer Overlay Networks

The capabilities of the Peer-to-Peer networks are limited. These overlay networks can be suitable to deploy many distributed applications and services but in some scenarios they may

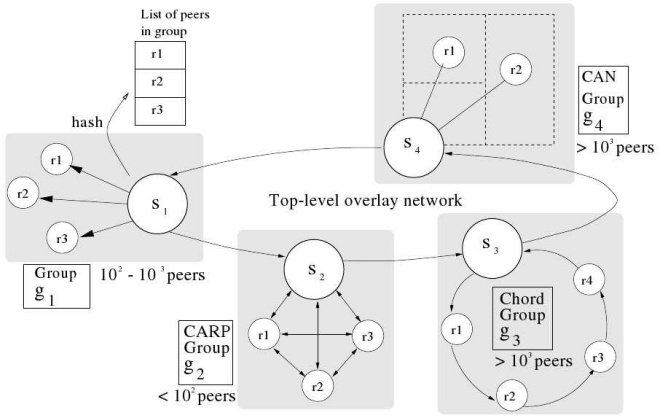


Figure 3.1: *Hierarchical Peer-to-Peer Overlay Network from [GEBR+03]*

not fulfil all the requirements. If the constraints for an application are very specific and they cannot be accomplished by a flat overlay network, it is necessary to explore new solutions. Some of these solutions are the usage of hierarchical Peer-to-Peer overlay networks.

Figure 3.1 illustrates an example of a Hierarchical Peer-to-Peer overlay network based on DHT's. This proposal was firstly proposed in [GEBR+03] and it consists in different overlay networks where each one runs its own overlay but they can exchange information through a top-level overlay network. The idea under this design is to take advantage of the heterogeneity of peers and take those ones with the best capabilities to play the role of super-peers. Furthermore, this design has other advantage: if the protocol used in an overlay becomes inefficient and it is necessary to change it, this change can be done with absolute transparency with respect the other overlay networks. Similar proposals to this idea have been realised [KF05]; however, anyone has proposed a common mechanism to easily define and maintain this kind of hierarchical overlay networks.

3.3 Different designs of Hierarchical Peer-to-Peer Overlay Networks

Since hierarchical overlay networks have been discussed in the scientific literature, many different proposals are published, each one focusing on some particular feature or optimization problem.

In [MS03a], [MS03b] a mechanism to improve file-sharing applications based on DHT's is proposed to form social groups. The idea is consider the meta-data information and type of files to group the items with the same characteristics in the same way the neighbours with the similar interests can be grouped together. In order to achieve this objective, the meta-data

information is embedded in each key in such a way that a kind of hierarchical ID is formed considering the groups previously defined.

Other approach is to minimize the delay of operation inside a Peer-to-Peer network. [XMH03] minimizes the delay in operations building a hierarchical overlay network. This hierarchical overlay network can be composed of several layers. Each peer is attached in all levels of the hierarchy. The levels of the hierarchy can be enumerated from 0 to n. Therefore, the smallest level of the hierarchy is 0 where different groups of peers are created; these groups are composed by those peers whose delay is the smallest among all the peers in the overlay. As bigger is the level in the hierarchy, bigger is the delay among the peers that populate the different groups in each level. The highest level contains all the peers in the overlay, so any peer can be reached if necessary. The routing works as follows: first, the routing is performed in the lowest level of the hierarchy; if the destination is not reached, the next level of the hierarchy is used. The idea is to use the fastest paths to reach the destination, however it implies an important overhead since each peer has to maintain several overlay routing tables in parallel, one per level. This design does not take into account the heterogeneity of peers and it can be problematic to hand-held devices with limited capabilities.

There are designs that also build a hierarchical architecture like [XMH03]. However, the objective is to improve the overall performance of the overlay without focusing only on the delay. [GGGM04] explains how a hierarchy can be built with a very simple merging algorithm. If two different overlay networks want to be merged (interconnected), additional links must be created between both overlay networks. These links are created as follows. Each node m in one overlay creates a link to a node m' in the other ring if and only if m' is one of the closest peers in someone of the routing entries pointing to some specific point in the space or if m' is the closer to m than any node in m's overlay. This mechanism implies to use the same overlay in the interconnected overlays since the usage of the metric in the link creation procedure makes necessary this requirement, in other cases this solution could not work.

[ALAS05] build a hierarchical DHT overlay network in a way very similar than [XMH03]; however, it assures that the number of links maintained by each peer is the same than the flat counterpart. The key space is divided in a prefix and suffix. The prefix is used in the own overlay and the suffix for the routing among clusters. This allows to take advantage of locality in most of the hops, only in the last ones is used other level of the hierarchy if necessary. Peers have to maintain their own overlay routing table; if the destination is reached, it is not necessary to do that any more. Nevertheless, if it is necessary to continue the search, other upper levels can be used. Therefore, other routing table is maintained with this purpose. This routing table has a tree structure, which is populated applying a XOR metric to the suffixes (pretty similar to Kademlia [MM02], but with a smaller number of routing entries). Furthermore, additional improvements can be applied if necessary [ALAS05].

Other approaches are different, focused only on one type of overlay network. For instance, [XZ02] and [ZWL07] build a hierarchical DHT overlay network based on CAN [RFH+01]. Basically, different clusters are formed dividing a virtual cartesian space in different contiguous subspaces. If a key does not belong to the subspace of a peer, the query is routed to the closest subset with respect to the query. Therefore, it is necessary to maintain

two routing tables, a routing table for the subset where the peer belongs and a routing table for the adjacent subsets. This can be done if it is assumed that each super-peer covers a virtual zone, which is equal to its subset, so the implementation is quite straightforward.

On the other hand, other designs like [ZDK06] study the cost of hierarchical DHT overlay networks, their behaviour and trade-offs. In order to perform this study, a cost must be assigned to the different actions performed in an overlay network, hierarchical or not. For instance, [ZDK06] considers the ratio between peers and super-peers and the number of exchanged messages in the overlay network and it studies how to minimize the number of exchanged messages. However, this methodology usually assumes some conditions in order to perform the analysis, i.e. churn is not considered in [ZDK06].

3.4 Super-peers management and trade-offs

Super-peers are peers that belong to a Peer-to-Peer overlay network, but they play a different role in comparison with legacy peers. They usually deploy some special functions that are necessary for the correct behaviour of the overlay networks. These functions are usually related with bootstrapping, storage of information (to reduce the amount of exchanged messages) or gateway functions. Therefore, the management of super-peers is really important in hierarchical Peer-to-Peer networks. Although some of them do not make use of them ([GGGM04], [XMH03], [ALAS05]), other ones need this role such as [XZ02], [ZWL07], [ZDK06]. Usually, super-peers take care about a subset of the entire overlay network, this subset is called cluster although it can also be also named as domain or group, (these nomenclatures would be used indistinctly along this Thesis). Next paragraphs consider the different mechanisms and trade-offs of using super-peers.

There is a lot of work related with the topic of super-peers in Peer-to-Peer networks. [BYGM03] explains how super-peers usually take the role of storage the index of the information available in an overlay network in order to centralize the query and avoid unnecessary traffic generated by the overlay. This fact is especially true if unstructured overlays are considered. Furthermore, it highlights how the topology created by the super-peers can be redundant or not. This is really important in critical operations: if a super-peer fails, the actions related with it could not be finished. Thus, it is important to have redundant super-peers to avoid unavailability problems with the services offered by them. Furthermore, it allows reducing the load of super-peers if load balancing is used [MCKS03]. In such a way, the bottlenecks are reduced and the probability of failure decreases. The open question that arises after this analysis is: which are the most suitable peers that can accomplish better the super-peer role?

One of the most important features of super-peers is their availability since their actions are necessary for the correct behaviour of overlay networks based on them. In [MHC06], a discussion about super-peer selection is provided. A mechanism to estimate the CPU usage is given, this mechanism is necessary if it is considered that super-peers usually need to perform more tasks than legacy peers do. Therefore, good super-peer candidates should have good and available processing capabilities that can be estimated according [MHC06].

Other parameters to select super-peers are considered in [MHC06], but they can be only applied to some file-sharing scenarios. The session and on-line time are also very important parameters for super-peer selection. If these parameters are large, it means that the super-peer is going to be available and this is a very desirable characteristic. Therefore, it is also interesting to take these parameters into account in the selection of a super-peer [LHMZ05].

In addition, if [BYGM03] is considered, we have to take also into account the bandwidth of super-peers since they have to support a larger load of messages. Thus, it is interesting to assure an enough quality in the available bandwidth of super-peers [ZWH03].

3.5 Conclusions

In this chapter, different kind of hierarchical overlay networks have been presented. Each one has been designed with very specific objectives and usually they are heavily coupled with a flat Peer-to-Peer overlay network, which has been taking as starting point to develop the hierarchical counterpart. However, there is not a proposal that allows setup different Peer-to-Peer overlay networks over a hierarchical architecture. For unstructured overlay networks, some super-peer selection mechanisms have been proposed but any global solution has been accepted. It is always mentioned that characteristics like bandwidth or processing power must be taken into account [BYGM03]. On the other hand, we have the structured overlay networks, which make difficult to define a mechanism to build hierarchical architectures because each Peer-to-Peer overlay network has a different structure. In some cases, such as [XMH03], [ZWL07] or [XZ02], the design is highly coupled to the flat Peer-to-Peer overlay network that is used as starting point. In other cases, the solution is more general such as [GGGM04] or [ALAS05] but the proposed mechanisms need certain modifications depending on the type of DHT overlay network that is chosen as starting point. Furthermore, all the individual overlays inside the hierarchical overlay network must support the same DHT but it would be desirable to have a greater flexibility inside each cluster. In relation with super-peers, few work related with structured Peer-to-Peer networks can be found [LHMZ05]. Therefore, it would be desirable to have a hierarchical design that allows the aggregation of different kinds of Peer-to-Peer overlay networks without taking into account their topology.

Los grandes conocimientos
engendran las grandes dudas.

Aristóteles (384 AC-322 AC)

Chapter 4

P2PSIP

This chapter contains an overview of P2PSIP Working Group and, in particular, its protocol RELOAD. P2PSIP is focused in the provision of a protocol that allows implementing any DHT overlay network and facilitates the deployment of Peer-to-Peer based systems with an open IETF standard. Thus, this work is really interesting for the objectives of this Thesis. It can offer an excellent tool to define a common payload and data structures to exchange information among the peers of different overlay networks with the advantages of a standard track.

4.1 Introduction

The IETF P2PSIP Working Group (P2PSIP WG) is standardising a protocol to support almost any Peer-to-Peer overlay network and it is attention is focused on DHT overlay networks. The current work is related with the development of a protocol named RELOAD [JLR+09a] and its main objective is the creation of an open standard to compete with Skype[1] [BS06]. This objective is motivated by the need of having a standard for developing Skype-like decentralised multimedia applications. In fact, the P2PSIP WG is chartered to develop protocols and mechanisms for the use of SIP [RSC+02] in environments where the service of establishing and managing sessions is mainly handled by a collection of intelligent end-points, rather than centralised SIP servers. However, the scope of P2PSIP is not limited to a distributed replacement of SIP by overriding the Proxy and Registrar SIP servers, but it could also be used for other purposes (for example file sharing or IPTV) or in combination with other signalling protocols.

Figure 4.1 presents the P2PSIP Overlay Reference Model using the basic concepts from [BMSW08]. P2PSIP protocol has to be designed to support any type of DHT overlay network. Each deployed overlay network is identified by an overlay name and the participants in this architecture can support two profiles: peers and clients. Peers are active

[1] http://www.skype.com

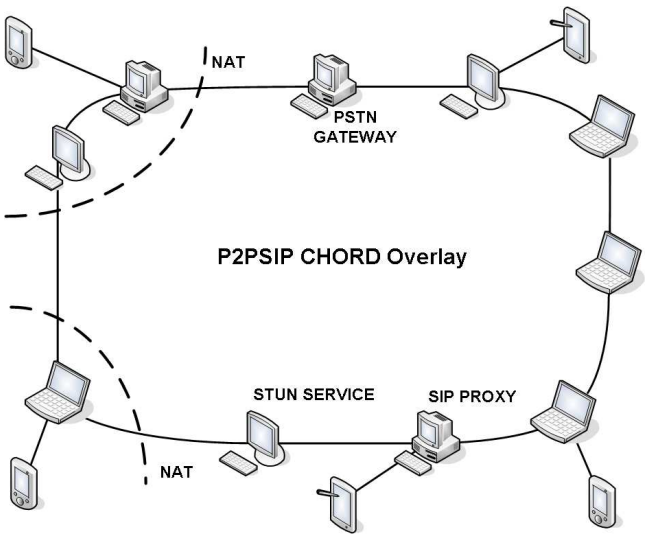


Figure 4.1: *P2PSIP Overlay Reference Model*

participant nodes in the overlay network and they are uniquely identified by a Node-ID (e.g. the computers and laptops in Figure 4.1). On the other hand, clients are entities that use the resources offered by the Peer-to-Peer overlay network but they do not participate in the overlay maintenance. This role is only reserved for devices with limited capabilities, such as the hand-held devices in Figure 4.1.

The information stored in the Peer-to-Peer network is made of resources where the information is placed. These resources are uniquely identified by a Resource-ID and restrictions do not exist about the information than can be stored in the overlay networks based on P2PSIP. In order to define the different type of data that can be handled in an overlay network, RELOAD [JLR+09a] defines the basic *kinds* (kind means data type in this context) that are associated with a Kind-ID. New kinds can be registered via IANA. In addition, private kinds can be defined if necessary for some scenarios.

Peers and services in an overlay network are identified according an Uniform Resource Identifier (URI). The format of this URI defined in P2PSIP is as follows: `RELOAD-URI="reload://"destination"@"overlay"/"[specifier]`. This P2PSIP URI defines RELOAD like the protocol that must be used to manage this kind of URI. In addition, the URI is composed by the destination (which can be a Node-ID or Resource-ID) and the overlay that is the Overlay-ID defined in RELOAD. However, the details of mapping the resources to a Resource-ID depend on each implementation (depending on the service to provide) and are independent of the functionalities offered by RELOAD. This mapping is open since offers a great flexibility to design any application based on RELOAD.

Another important point in RELOAD is the fact that the protocol must support basic primitives for any Peer-to-Peer overlay network such as joining, bootstrapping, resource allocation and maintenance. Furthermore, it must allow the connectivity among the peers in the overlay network even if peers are behind NAT's when any of the previous primitives is

used.

It must be considered that P2PSIP WG re-implements in RELOAD the proxy and registrar entities of SIP in a decentralised fashion but it can also be used for other purposes. The user and service information is distributed among all peers in the Peer-to-Peer overlay network, instead of storing it in the registrar and proxy servers. The requests for this information are also handled by the overlay infrastructure. The advantages of P2PSIP include the elimination of single points of failure (because of its decentralised nature) and it reduces the bandwidth costs for Service Providers (SP's). Finally, to support the interoperability between P2PSIP and conventional SIP entities, special services and capabilities must be provided [MB07].

Based on the requirements for the P2PSIP protocol previously presented, the RELOAD protocol [JLR+09a] has been proposed as working group draft. One of the most relevant decisions is the adoption of a binary protocol instead of a character based protocol, resulting in a lightweight protocol suitable for peers that have to manage a lot of connections and resources (CPU, bandwidth, etc). The protocol is based on a modular design (see Figure 4.2) that supports different overlay networks and can be used by any application. In fact, several proposals for different usages are being provided by the IETF community. These usages go from the original SIP replacement [JLR+09b], [GM09] to file sharing [WSM09] or IPTV support [KKO+09].

The transport protocols used in RELOAD are TLS [DR06] and DTLS [RM06], connection-oriented and connectionless respectively. Both of them provide secure data flows and their selection would depend on the necessities of the application and the overlay network that would be used.

4.2 Architecture

RELOAD has a modular design (Figure 4.2) that allows the differentiation between the packet format and the different modules that work together to obtain the desired functionality. The Topology plug-in is responsible of implementing the DHT overlay algorithm. This one is connected with the Message Transport module, which handles the end-to-end reliability of any exchanged message. The Storage module handles the storing of resources in the overlay network and it is connected with the Topology plug-in that knows how the overlay network is organized and the replication policy that depends on the implemented overlay network. Furthermore, it is connected with the Message Transport module since it handles the transmission of Fetch operations and communicates the results to the Storage module to properly manage the different petitions. Finally, a Forwarding layer delivers the messages using the Interactive Connectivity Establishment (ICE) [Ros07] protocol based on STUN and TURN servers to cross NAT's if necessary. An additional connection between the Forwarding layer and the Topology plug-in is used by the Forwarding layer to notify when a peer is not reachable; this information can trigger maintenance operations in the Topology plug-in such as updating the routing table. In fact, a great effort is being invested by the WG to have a good set of diagnosing tools [SJEB09]. These tools can be used to detect problems

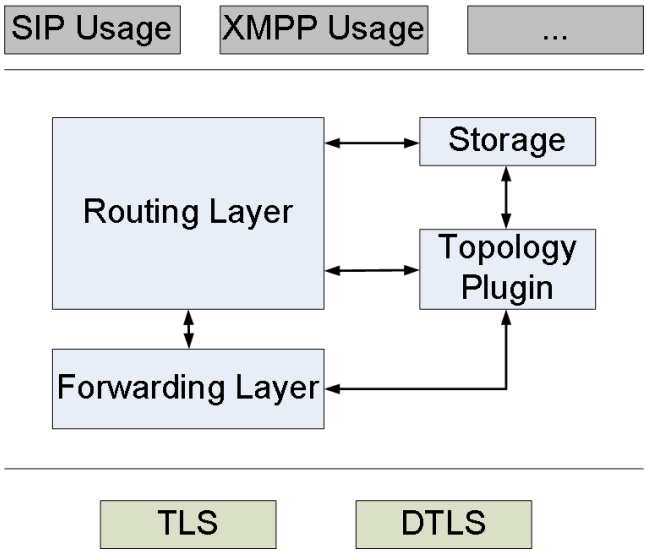


Figure 4.2: *P2PSIP protocol reference model*

in the overlay network, which must be managed by the topology plug-in or for debugging purposes when a new topology plug-in is being developed since it needs a detailed test and validation in real environments.

Due to compatibility reasons, it is mandatory to implement a Topology plug-in in any RELOAD implementation. This implementation is based on Chord [SMLN+03] and it is specified on the protocol definition document [JLR+09a]. However, it has been tested that in real conditions its functionality is limited. The design is the main problem of Peer-to-Peer applications. Many problems arise when an implementation is tested under a real environment, which is more complex than any simulation or small testbed. One problem in a real implementation is the changing number of users (peers) that are involved in the overlay network. Depending on this number, some designs can be more suitable than others can. One solution can be to set up a dynamic adjustment according the estimations of some parameters in the overlay network. Some information in relation with this topic can be found in [MCH09]. However, depending on the type of overlay network, the adjustment parameters can be different. Due to this work, new topology plug-ins are in development to improve the overall performance [MSD+09]. Nevertheless, there are continuous contributions in this field and this is one of the main advantages of RELOAD. Since it has a modular design, any Topology plug-in can be designed and implemented to fulfil any desired requirement if necessary.

4.3 Message Handling

The performance of an overlay network is closely related to the Topology plug-in and the Message Transport module. The messages used in a Peer-to-Peer overlay network can be transmitted in an iterative (the peer that performs the query takes care of each hop) or

recursive mode (the peer in each hop takes care of the next hop). Depending on the scenario, an option would be more suitable than the other one. In general, the recursive mode is preferred because in most cases it would get a lower delay; this fact is also closely related with the requirement of supporting NAT in a transparent way. When recursive routing is used, a peer forwards a message to the next hop according to its overlay routing table. If those peers are in the overlay routing table, they must have been contacted previously using ICE for NAT traversal if necessary. Therefore, if a message must be forwarded because the recursive mode is being used, it would not be necessary to perform any ICE exchange since it was done previously and each peer has a cache for this purpose. On the other hand, if iterative routing is used, most probably the next hop is not known by the peer performing the operation since the desired entries are not available in the cache. Therefore, an ICE exchange is necessary per each unknown next hop, which implies an undesirable impact on the delay.

The following components are used to route queries inside an overlay network in RELOAD. The first is the Node-ID, currently with 128 bits but a variable length field would be more useful and we advocate for this option (details are in chapter 6). Resource-ID's are expected to have a variable maximum length of 255 bytes. If the Resource-ID is longer in length than the Node-ID, then it should be truncated to the Node-ID length for storage and fetch operations. The overlay messages contain two additional data structures: the Destination-List and the Via-List. The Destination-List allows specifying a list of intermediate peers and it can be used to avoid unnecessary ICE exchanges, these peers are named relays. The Via-List is used to get a response path symmetric to the request path (Figure 4.3) and it is the default mechanism in RELOAD [JLR+09a]. Another option would be that the contact info of the peer sending the message is included to allow a direct response (Figure 4.4) or through a relay if it is behind a NAT. These last two mechanisms are covered in [JEB09]. Although they seem to be more efficient from the point of view of delay, this is not necessarily true. The direct response and the usage of a relay imply a smaller number of hops. However, the total delay depends not only on the sum of the delay in each hop, but also on the time needed for ICE exchanges to discover an available pair of locators (IP addresses) between each hop if these ones are not available directly. If the Via-List mechanism is used, higher is the probability of not performing an ICE exchange in responses. This fact is because the information of the valid IP addresses along the path should be previously cached in advance with high probability when the queries are forwarded to their destination the first time.

Regardless of the mechanism used on the return path, once the information is retrieved, the next step depends exclusively on the application level. If multimedia applications are being developed, the end-points can proceed with the establishment of the multimedia session. For this scenario, Figure 4.3 and Figure 4.4 show a SIP exchange where the negotiation of these session parameters is performed. For other types of applications, the underlying SIP exchange is replaced by another suitable protocol.

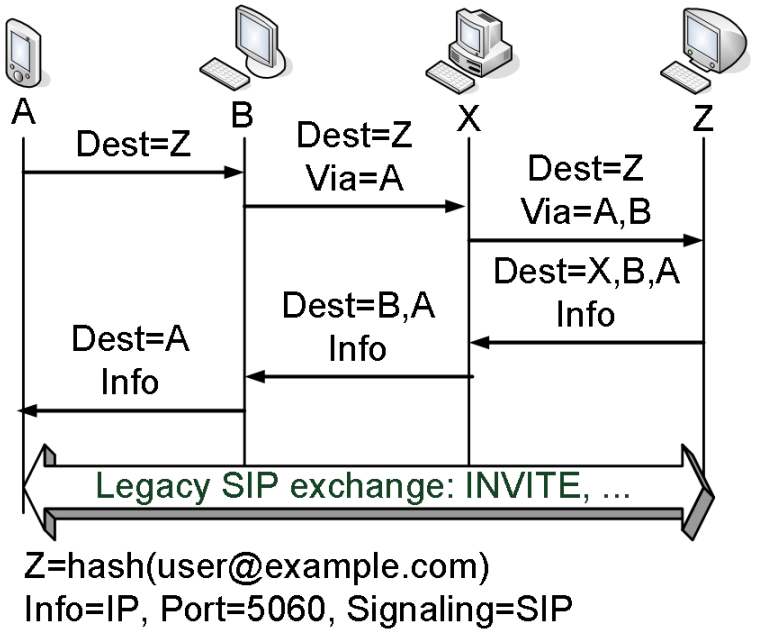


Figure 4.3: *Reload using Via-List*

4.4 Signalling

RELOAD has defined some primitives, which allow the operation of almost any DHT overlay network. These primitives allow the exchange of information among peers with various objectives: maintenance of the state in the structured overlay network, storage of the information in the overlay network and retrieval of the information from the overlay network. These primitives are Join, Leave, Store, Fetch, Remove and Update. In addition to these ones, there are other operations for debugging and testing.

4.4.1 Join

A new peer uses the Join primitive to join the overlay. The Join is sent to the responsible peer depending on the routing mechanism described in the topology plug-in. This notifies the responsible peer that the new peer is taking over some of the overlay and it needs to synchronize its state. If the request succeeds, the responding peer must follow up by executing the right sequence of Stores and Updates to transfer the appropriate section of the overlay space to the joining peer.

4.4.2 Leave

The Leave message is used to indicate that a node is exiting the overlay. A node should send this message to each peer with which it is directly connected prior to exiting the overlay and distribute the information managed by the peer among its neighbours. Upon re-

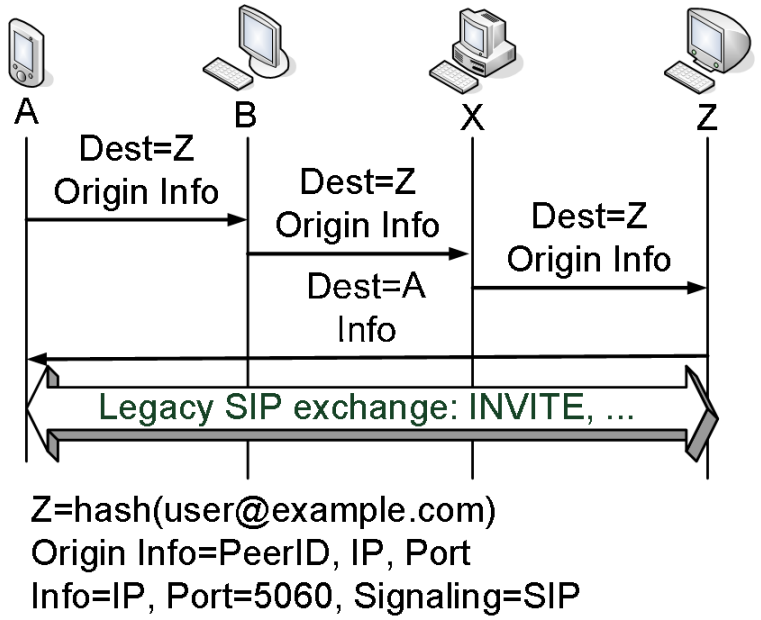


Figure 4.4: *Reload Direct Response (optional)*

ceiving a Leave request, a peer must update its own routing table, and send the appropriate Store/Update sequences to re-stabilize the overlay.

4.4.3 Update

Update is the primary overlay-specific maintenance message. It is used by the sender to notify the recipient of the sender's view of the current state of the overlay (its routing state) and it is up to the recipient to take whatever actions are appropriate to deal with the state change.

4.4.4 Store

The Store method is used to store data in the overlay. Depending on the topology plug-in, some mechanisms for the storage of redundant replicas must be specified. In any case, the following actions are necessary:

- The data type of the information stored must be supported by the overlay network.
- The data type of the item is exactly the defined in data field of the Store primitive (consistency).
- The signatures over each individual data element (if any) are valid.
- Each element is signed by a credential, which is authorized to write this type at this Resource-ID.

4.4.5 Fetch

The Fetch request retrieves one or more data elements stored at a given Resource-ID. A single Fetch request can retrieve multiple different items.

4.4.6 Remove

The Remove request is used to remove a stored element or elements from the storing peer. Before processing the Remove request, the peer must perform the following checks:

- The data type must be supported.
- The signature over the message is valid or (depending on overlay policy) no signature is required.
- The signer of the message has permissions, which permit him to remove this type of data. Although each data defines its own access control requirements, in general only the original signer of the data should be allowed to remove it.

4.4.7 Example

An example of the signalling in RELOAD is illustrated in Figure 4.5. This figure shows the legacy operation of storage and retrieval in a VoIP scenario in order to give a probably scenario of usage. A peer can store its IP location using the store primitive, the key where this information is stored is associated to hash of its SIP URI. In RELOAD, any operation is acknowledged and a Response is sent back. Later, a peer in the overlay wants to retrieve that information so it uses a Fetch primitive, which contains the hash of the SIP URI that wants to be retrieved. Once this petition arrives at the destination, the query is answered with the associated information. For simplicity, the intermediate hops which must follow the different petitions have not been included in the picture; however they must follow the handling in each hop as it has been explained in section 4.3. The associated responses should follow the Via-List mechanism (Figure 4.3) although Direct Response (Figure 4.4) could be also allowed if necessary.

Other actions like Join or Leave operations are tied to the overlay topology and they are associated with Store and Remove operations, which are executed depending on the kind of overlay and the chosen configuration parameters of the overlay. Thus, they are not explained, however additional details can be found in [JLR+09a].

4.5 Conclusions

This chapter explains the ongoing work in the P2PSIP group about defining a new protocol that allows the implementation of any DHT. This protocol is named RELOAD and is

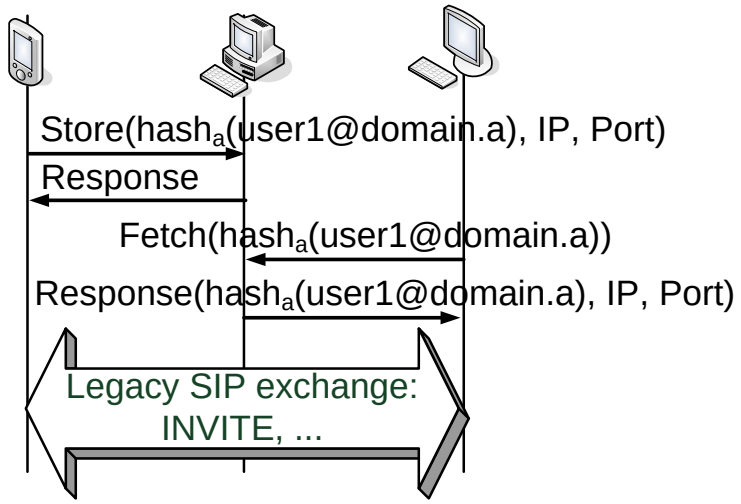


Figure 4.5: *Signalling exchange*

based on a modular infrastructure. This modular infrastructure decouples as much as possibility the Peer-to-Peer overlay functionality from the other goals of the protocol such as NAT Traversal, replication or diagnostics. Furthermore, it defines the basic primitives that must support any DHT. In addition, special attention has been considered in designing different mechanisms to route the traffic in order to optimize the performance. These optimizations take care about the NAT traversal scenario where the shortest path is not the fastest one. Finally, the signalling associated to this protocol is presented.

The efforts of the P2PSIP WG and the ongoing design RELOAD offers a great tool to implement any DHT overlay network. Therefore, it is an excellent candidate to complement its design with some additions that allow the exchange of information among different overlay networks with a common payload and data structures. This last fact is really interesting if we consider that is one of the objectives of this Thesis.

Part II

Design of interconnecting structured Peer-to-Peer networks with a hierarchical architecture

Ciencia es todo aquello sobre lo
cual siempre cabe discusión.

José Ortega y Gasset (1883-1955)

Chapter 5

Goals and design considerations

This chapter summarises the concepts and open issues identified in previous chapters and looks for new proposals in the research topic of Peer-to-Peer overlay networks and hierarchical Peer-to-Peer overlay networks. In addition, it proposes the goals and objectives of this Thesis as well as the adopted methodology that assures the quality of the obtained results in this work.

5.1 Introduction

In previous chapters, an overview of the state of the art in Peer-to-Peer and Hierarchical Peer-to-Peer networks is detailed. It is demonstrated that Peer-to-Peer networks are a good solution for distributed environments but their maintenance and management tasks are not trivial to support. Considering the great number of application scenarios, not perfect solution exists, which solves any problem in a completely distributed fashion. A trade-off among different parameters always exists and depending on the goals (performance, speed, monetary cost, etc); these parameters are selected in a different way.

In addition, in order to improve the capabilities of traditional Peer-to-Peer networks, hierarchical architectures have been proposed in the literature. Their objective is usually to improve the performance of their flat counterparts; this objective is normally obtained by reducing the delay in the Peer-to-Peer overlay operations or minimizing the costs of maintaining the overlay structure that supports the distributed application. However, these solutions usually are heavy coupled to the original flat Peer-to-Peer network and they are not easy to extrapolate to other overlay networks. Therefore, their extensibility and applicability are only valid in a small area of work. Furthermore, several solutions can solve the same problem, perhaps with slight differences, but the interoperability with these solutions is in general not possible. Thus, a common framework to support the interaction among different Peer-to-Peer overlay networks is desirable in any case; however, no solid proposal has been presented until now. Therefore, the main goal for this Thesis consists in defining a

hierarchical Peer-to-Peer architecture that allows the exchange of resources among different Peer-to-Peer overlay networks.

5.2 Goals

5.2.1 General Hierarchical DHT Overlay Network

This goal consists in providing a solution that allows the creation of hierarchical DHT overlay networks in an easy and compatible way. This means to provide a mechanism that allows the extension of flat DHT solutions to a hierarchical counterpart avoiding the redesign of certain parts of the original flat proposal. Furthermore, the proposed solution must allow creating a hierarchical DHT network in such a way that *different* DHT overlay networks can be used in the hierarchical topology if necessary.

5.2.2 Availability for exchanging information among different overlays

This objective is complementary to the previous one. Right now, different overlay networks cannot exchange information among them; each overlay defines its own mechanism for the exchange of information which does not allows any kind of interoperability among different Peer-to-Peer overlay networks. Therefore, it would be really interesting to allow different overlay networks to exchange their resources. The idea is to make use of a hierarchical Peer-to-Peer network in order to allow this communication and exchange of information.

5.2.3 Use of open standards

The communication and exchange of information among different Peer-to-Peer overlay networks in a general hierarchical Peer-to-Peer communication is not a trivial issue that should be generally accepted. Therefore, some mechanism must be introduced with this purpose. It would be desirable to use open standards that assure an open specification that different implementations can easily follow. In such a way, it can be assured that the deployment of different overlay implementations ends in a real inter-operative deployment.

5.2.4 Provision of mechanisms to assure the scalability of the solution

To allow the exchange of information among different Peer-to-Peer overlay networks is not a trivial issue. Therefore, it would be necessary to introduce new mechanisms to allow the desired interoperability. However, the design has to assure the scalability of the solution since it is the strong point of Peer-to-Peer overlay networks. In fact, the design should have at least the same scalability than a flat overlay network if possible.

5.2.5 Validation of proposed solutions

In order to assure the quality and efficiency of the proposed design, it is important to validate our solution. We adopt a validation process that consists in three steps:

1. A mathematical analysis to assure that the proposal is reasonable and can give interesting results.
2. A simulation that assures that the mathematical analysis is valid.
3. A real implementation that removes most of the previous assumptions in order to assure that the proposed solution can work in real environments. The implementation could show problems of interaction when a real TCP/IP stack is used if someone exists.

5.3 Conclusions

This chapter presents the goals and methodology that is going to be followed for the research in this Thesis. These goals are focused on providing a general and flexible solution that allows its adoption in almost any scenario if necessary. On the other hand, the defined methodology follows the traditional flow of design, implementation (in a simulator or a real application) and experimentation. These steps are necessary to assure that our work covers with scientific rigour all of our initial objectives. If these steps are strictly followed and the obtained results are relevant and in the range of our expectations, it could be concluded that the objectives have been fulfilled and the Thesis is finalized.

La religión sin la ciencia estaría ciega, y la ciencia sin la religión estaría coja también.

Albert Einstein (1879-1955)

Chapter 6

Hierarchical P2PSIP (H-P2PSIP)

This chapter presents a general hierarchical DHT architecture that allows the exchange of information among different DHT networks regardless of their topology or structure. Several key points compose this architecture. First, a Hierarchical-ID is used to route the queries among different domains (each domain implements its own overlay). Second, this architecture makes use of super-peers, which route the queries among different domains. Third, the P2PSIP protocol (RELOAD) is adopted to assure the exchange of information among peers of different domains. Finally, a signalling procedure is defined to allow an ordered exchange of information among different overlay networks.

6.1 Introduction

The idea of a H-P2PSIP is based on Figure 6.1 where a hierarchical overlay network interconnects different types of overlay networks. The work related with this topic can be partially found in [MYBG+08a], [MYBC+09]. Basically, two actions are needed to define an architecture that supports the exchange of information among different overlays. First of all, a common protocol is needed that allows the implementation of any desired Peer-to-Peer overlay network, which can be used by any Peer-to-Peer application based on it. The second point consists in a mechanism to build a hierarchical architecture in such a way that its definition, management and maintenance allows the usage of *any DHT* overlay network. Therefore, the most suitable DHT overlay network could be selected depending on the scenario.

As the name of our proposal suggests, H-P2PSIP, the protocol selected to support the desired hierarchical Peer-to-Peer architecture is the P2PSIP protocol. Although RELOAD [JLR+09a] has been explained in chapter 4, as summary it can be said that RELOAD provides the flexibility to support any DHT overlay network. Although the standardization process is not concluded, RELOAD already shows its utility and benefits. Previous designs such as P2PP [BSM07] demonstrate that it is possible to have a common protocol to deploy

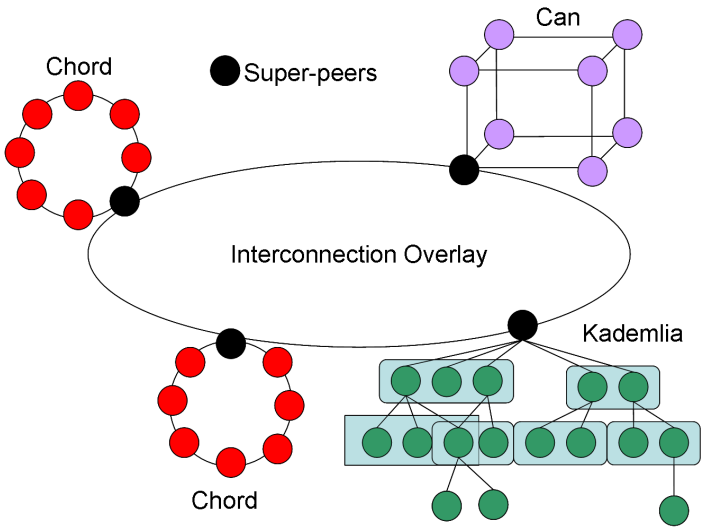


Figure 6.1: *H-P2PSIP overview*

different Peer-to-Peer networks. P2PP can be considered as a precursor of RELOAD and some implementations [1] [Coh08] are available supporting this concept.

Nowadays, there is not a strong interest in supporting hierarchical Peer-to-Peer overlay networks in P2PSIP yet. Although some people are pointing attention to this issue [Le09] or [Hua09], it is not succeeding in the P2PSIP WG since this WG is focused on defining an operational protocol for legacy DHT overlay networks. However, this must not be considered like an uninteresting topic because the IETF tries to define standard tracks and it does not consider research topics. The contribution of this Thesis is focused in providing the mechanisms, as simple as possible, to support a H-P2PSIP architecture in such a way that allows the exchange of information among different DHT overlay networks with a hierarchical architecture (see Figure 6.1).

6.2 H-P2PSIP architecture

H-P2PSIP defines a hierarchical overlay network composed of two levels of hierarchy, an example is given in Figure 6.1. The lower level is populated with different domains that want to exchange information among them. Each domain is independent with respect to the others; therefore, each domain implements its own overlay network (Chord [SMLN+03], Kademlia [MM02], Can [RFH+01], etc) according to the preferences in each domain. On the other hand, the upper level is composed only by an overlay network named Interconnection Overlay. This Interconnection Overlay acts like a directory service among the different domains in the lower level of the hierarchy. Its purpose is to route the queries among different overlay networks when a peer of one domain wants to retrieve information placed in other

[1] http://www1.cs.columbia.edu/ salman/peer/

domain. This Interconnection Overlay can be implemented using any desired overlay network. The purpose of this two level hierarchy is the exchange of information among different overlays. However, some open question remains:

- How are the resources stored in this hierarchical architecture?
- How is the information is routed among different overlay networks?
- How can peers in different overlay networks exchange information among them?

These questions are answered in the next sections of this chapter.

6.3 Hierarchical space domain of identifiers

In order to support the H-P2PSIP architecture, we define a hierarchical space of identifiers composed of Hierarchical-ID's (see Figure 6.2). A Hierarchical-ID is composed by two concatenated ID's: a Prefix-ID with n bits and a Suffix-ID with m bits.

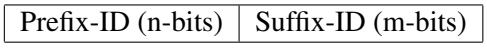


Figure 6.2: *Hierarchical-ID*

The Prefix-ID is used for the routing queries in the Interconnection Overlay among the different domains. This implies that all the peers or resources belonging to the same domain share the same Prefix-ID. On the other hand, the Suffix-ID is used only in the domain where a peer is attached and it permits to localize any resource in the overlay network of that domain. Thus, this design allows the routing of queries among different domains. When looking for a resource in another domain, the query is routed to the desired domain using the Prefix-ID. Finally, the desired resource in the external domain is found through the Suffix-ID.

6.3.1 Hierarchical-ID generation

The RELOAD specification it does not define how the Node-ID's or Resource-ID's are generated, however it is convenient to define some mechanism that allows implementing easily any desired service. It must be considered that a lot of information in the Internet is identified with an URI. Therefore, it would be interesting to generate the Hierarchical-ID's based on regular URI's. In the next paragraphs is explained how the Node-ID's and Resource-ID's can be generated applying has functions to a regular URI. The starting point for this discussion is that each domain has a domain name (i.e. `example.com`) and its resources are identified by an URI (i.e. `resource@example.com`).

6.3.1.1 Node-ID generation

A Node-ID identifies each node participating in the overlay network. How this Node-ID is generated depends on the security level desired in the system. If we have a domain named `examle.com`, the Prefix-ID can be generated as follows: `Prefix-ID=hash(example.com)`. This construction of the Prefix-ID allows the generation of any desired Prefix-ID if the domain that wants to be contacted is known in advance. On the other hand, the Suffix-ID must be different for each peer. One option is `Suffix-ID=hash(ip_address)`, but any other unique characteristic of the peers can be used. The inconvenience of generating the Suffix-ID in such way is the fact that this method is insecure since Sybil attacks can be performed [UPvS09] and it presents problems with NAT's. However, if a more secured infrastructure wants to be provided, a central authority per domain can be used to generate certificates for the peers in that domain [UPvS09]. The certificates from the central authority, in addition to the Prefix-ID, contain the Suffix-ID of peers based on a random number. Thus, the Sybil attacks are avoided in this way and the problems with NAT's are solved.

6.3.1.2 Resource-ID generation

If we have a resource associated to a regular URI like `resource@example.com`, a Hierarchical-ID can be created using this URI. The Prefix-ID in a Resource-ID must have the same value used in the Node-ID's. In fact, we can do this since the URI contains the domain where the resources are. Therefore, the Prefix-ID must be: `Prefix-ID = hash(example.com)`. On the other hand, the Suffix-ID must be different for each resource if possible; thus, the Suffix-ID can be generated hashing the whole URI: `Suffix-ID = hash_a(resource@example.com)`. The hash functions `hash` and `hash_a` can be identical or different.

Once the mapping between the URI's and Hierarchical-ID's is established, any resource can be stored with its Resource-ID and the original URI in the peer with the closest Node-ID. Depending on the DHT protocol, this information can be replicated to other peers if necessary. Thus, any content associated with a regular URI can be stored in a DHT overlay network as a resource. The content of the resource can vary depending on the application scenario. In the case of a VoIP application, it can be the user contact information, supported protocols and codecs. In the case of some service, it can be its configuration parameters.

6.4 P2PSIP considerations

In our Hierarchical-ID proposal, a flexible length for the Prefix-ID and Suffix-ID is suggested. They can have the same length or not, it depends on the used hash functions and how the result of these hash functions is truncated. This Hierarchical-ID is used in Node-ID's and Resource-ID's and they have already been defined in RELOAD [JLR+09a]. The Node-ID in RELOAD has 128 bits whereas the Resource-ID has 255 octets. In RELOAD,

if the length of the Resource-ID is larger than the Node-ID, the Resource-ID is truncated to the Node-ID for storage, retrieval and routing purposes. If our Hierarchical-ID wants to be used with RELOAD, there is not any problem with respect to the Resource-ID's; the length in RELOAD for these Resource-ID's is long enough. On the other hand, the Node-ID has a fixed length of only 128 bits and this is a limitation. One option is to make the sum of the Prefix-ID and Suffix-ID equals to a length of 128 bits. This solution probably would work in most of the cases. However, if the number of nodes is very large, a length of 128 bits could be problematic. Therefore, in order to avoid future problems, it would be desirable, if possible, to make the Node-ID in RELOAD with a variable length like the Resource-ID's. This point is being currently in the P2PSIP WG.

6.5 Super-Peer role in H-P2PSIP

Super-peers are necessary to interconnect the different domains and to route the queries among these domains, see Figure 6.1. It is mentioned before how the routing is based on the defined Hierarchical-ID but the mechanism to make use of this ID has not been specified. The idea is not to overload most of the peers with additional tasks, so we use super-peers to realize the necessary additional operations. Basically, super-peers have to forward the queries to the right external domain specified in the query. Thus, the super-peers must participate in the Interconnection Overlay, as well as in their own domain as regular peers. Taking into account the role of super-peers, their main tasks are as follows:

1. Enrolment and maintenance operations in the Interconnection Overlay.
2. Forwarding of inter-domain queries.

The first task corresponds to building and maintaining the Interconnection Overlay. In this overlay network, all the super-peers are attached and they publish their information. Among this information two important parameters are published, the location information of each super-peer and the domain that each super-peer is taking care. Therefore, if a peer receives a query for another domain, it looks for the super-peers that are taking care of that domain and retrieves their location information to forward the previously received query. Since super-peers have to manage the Interconnection Overlay, they have to manage an additional overlay routing table in addition to the regular one that they need to perform queries inside its own domain.

The second task is associated with the forwarding of queries to other domains. This task implies a larger load in terms of CPU consumption as well as bandwidth requirements in the super-peers. This fact makes necessary to choose the super-peers carefully and consider the heterogeneity of peers to select the most suitable ones. In the literature, there are mechanisms to select these super-peers in a proper way [MHC06], [MCKS03]. These mechanisms can be integrated in the maintenance protocol of the DHT used in each domain. Each domain must have at least one super-peer although it is desirable to have several super-peers for redundancy and load distribution.

6.6 Query generation in H-P2PSIP

If a query belongs to the own domain of a peer (intra-domain query), it is not necessary to perform any modification to the search process. However, if a query has its destination in another domain (inter-domain query), several things must be taken into account if a successful response is desired. We have to highlight that any hash function can be used to map the URI of the resources to the Resource-ID. The hash functions are not two-way functions and an URI cannot be recovered from a Hierarchical-ID. In fact, several URI's could have the same Hierarchical-ID although the probability of collision is small (depending on the size of the ID). Therefore, some mechanism must be provided to avoid this problem. If a common mapping function is used to map the URI's in Hierarchical-ID using a hash function, the best way to proceed is to send, in conjunction with the Hierarchical-ID, the URI of the resource. In this way, the generation of the Hierarchical-ID can be recalculated according to the rules in the destination domain and the retrieval of the resource information could be done in any case.

6.7 H-P2PSIP Signalling

The H-P2PSIP signalling is associated to all the operations in the overlay network but the most interesting operations are the storage and retrieval of the resources in the overlay network. After resources have been mapped to identifiers and a criterion for their storage has been defined in the Topology Plug-in, any resource can be stored in the overlay. This storage is the same defined in P2PSIP since the storage of resources is placed in the own overlay network, only searches are performed outside the Peer-to-Peer overlay network if a resource from other domain wants to be obtained. Therefore, the main open issue is the retrieval of information. We have two different cases that must be considered:

1. Intra-domain queries: the search of a resource is bounded to the P2PSIP domain of the requester. This case is really simple since the search for resources is done inside the P2PSIP domain and it is identical to the flat Peer-to-Peer overlay using only the Suffix-ID. In this situation, the Prefix-ID of the resource must be equal to the hash of the associated URI domain. This hash is known by all peers belonging to the same P2PSIP domain.

2. Inter-domain queries: the queries look for a resource that is placed in a different domain. Since all peers in a domain know at least one super-peer, they can send a query to the super-peer in one hop. When the super-peer receives the query, it will search in the Interconnection Overlay for any of the super-peers that are responsible for the targeted Prefix-ID and once this information is retrieved, the query is forwarded to one of these super-peers. When the super-peer of the destination P2PSIP domain receives the query, it forwards the query inside its domain. If the query reaches a peer that has the desired resource, then the peer replies in such a way that is compliant with the P2PSIP protocol [JLR+09a].

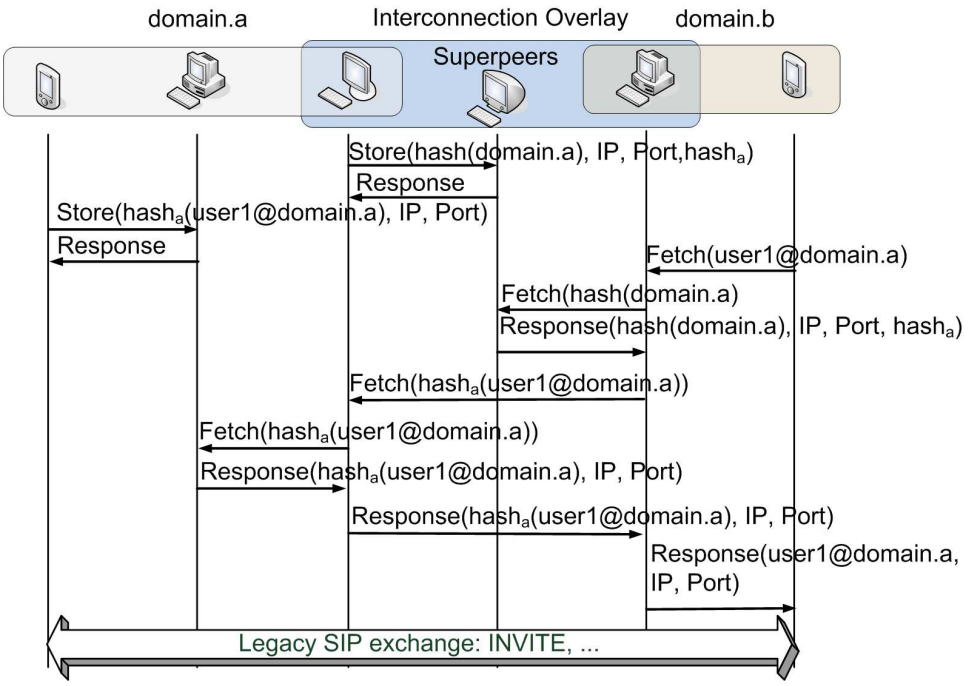


Figure 6.3: *H-P2PSIP Signalling*

An example of the signalling on the proposed hierarchical scenario is shown in Figure 6.3. Several aspects are taken into account in order to understand the signalling flow. The peer in `domain.a` performs the storage operation according to the definition in the RELOAD specification. Later, peer in `domain.b` requests the information of `user1@domain.a`, the query in the Fetch message is plain text since the Hierarchical-ID can be rebuilt with independence of the hash function used in the different overlays if the mapping function is correctly predefined. Thus, once the super-peer at `domain.b` receives the query, it performs a query of `hash(domain.a)` in order to obtain the information related with the super-peers in `domain.a` through the Interconnection Overlay. Inside this information, the hash used in the other domain ($hash_a$) is included and a request for the desired item can be built as `hash_a(user1@domain.a)` and sent to the super-peer in `domain.a`. The super-peer from domain `domain.a` forwards the information to its overlay and waits to get the desired answer. Once the answer is received, the super-peer from `domain.a` forwards the resource information to the super-peer from `domain.b`. Finally, the super-peer from `domain.b` sends the response to the peer from `domain.b`, which realised the original query. The answer follows the reverse path using the Via-list to follow the default behaviour in RELOAD. For instance, if the requested information it was the location information associated to a SIP URI, a SIP negotiation can be initiated for IM, VoIP or Video Conference with the received information. Figure 6.3 illustrates a subset of the real flow. The figure omits the intermediate hops in each overlay and ICE exchanges, if any is needed.

6.8 Characteristics of the H-P2PSIP architecture

The H-P2PSIP proposal has several advantages. In order to see these advantages, in the next paragraphs, a comparison with the flat overlay networks counterpart is made since they are a good point of reference, which can be used to compare our proposal. First, the operations or primitives of the DHT used in H-P2PSIP are not modified. Only some changes are needed in the maintenance operations to include the selection and update of super-peers [MHC06], [MCKS03], which can be included as options in RELOAD. Furthermore, the routing state in legacy peers does not increase compared to a flat overlay network because the number of maintained peers per overlay network does not increase, it is maintained to the number of peers in each P2PSIP domain. Hence, the number of the routing entries is limited by the number of peers in each domain, although connectivity with other P2PSIP domains is available, which was a prerequisite in the design rules. If we consider that the Routing State in a Peer-to-Peer network usually depends on the logarithm of the number of peers [LCP+05], we have that the Routing State in our approach is $O(\log_B M)$ where M is the number of peers in a domain. If we compare this Routing State with a unique flat P2PSIP domain that contains all P2PSIP domains, we obtain that the number of peers in the flat overlay network is $M \cdot K$ where K is the number of domains. Thus, the Routing State is increased up to $O(\log_B(M \cdot K))$. Thus, using the hierarchical architecture, legacy peers save $log_B(K)/log_B(M \cdot K) \cdot 100\ \%$ of overlay routing entries in comparison with the flat counterpart. In addition, the Routing State is independent with respect the number of P2PSIP domains, which is also an interesting property.

Other approaches like [XMH03] or [GGGM04] propose more complicated hierarchical architectures in order to obtain overlay network with short delays, but their solutions also imply an increment of the Routing State, which is not suitable for our scenario since hand-held devices are considered to provide VoIP communications. In the hierarchical case, we show how the Routing State is reduced for the same number of peers but a comparable Routing Performance is maintained.

The drawback of this approach is the overload of super-peers [BYGM03]. Nevertheless, this overload is smaller than in other proposals [GEBR+03], [ZDK06] where the super-peers must store the information of all peers that depend on them. This fact implies the maintenance of a larger amount of information with respect to the peers. Actually, our super-peers have to maintain two routing tables: a routing table of size $O(\log_B M)$ for their own domain and a routing table of size $O(\log_B K)$ for the Interconnection Overlay. In this case, the Routing State in super-peers is $O(\log_B M)+O(\log_B K) = O(\log_B(M \cdot K))$, this values corresponds with an equivalent flat overlay network. This Routing State is the same one found in [GGGM04] and smaller with respect to [XMH03] but with an important difference: the Routing State in the previously mentioned proposals is maintained by all peers in the hierarchical overlay network and we only need to maintain this Routing State in the super-peers. Nevertheless, the real cost in our super-peers is the need of additional bandwidth and CPU processing to forward the queries among the different domains. Considering the heterogeneity of peers, this fact is not a problem if the most powerful peers are selected as super-peers.

6.9 Application scenarios

In this section, some application scenarios are explained demonstrating the utility of a H-P2PSIP architecture.

6.9.1 VoIP

This scenario has been vaguely mentioned during the explanation of H-P2PSIP but it is necessary to explain it in detail. A real application of H-P2PSIP is VoIP; in fact, applications like Skype are really popular and an open standard application would be desirable especially if can be used for SP's and ISP's. Nowadays, ISP's can offer VoIP services based on SIP. However, the support of SIP implies a centralize entities as the Proxy and Registrar servers and their maintenance cost increases with the number of users. Therefore, a less cost effective solution is desired and this solution can be H-P2PSIP. Each ISP can deploy its own P2PSIP domain for VoIP. In the Peer-to-Peer overlay network are stored the SIP URI's of the users according to a predefined mapping function. Therefore, the location information associated with SIP URI's can be stored with its associated Resource-ID. If a conference wants to be maintained with a user in other domain, the super-peers in H-P2PSIP allow retrieving the desired information in other domain to open a VoIP call. If we consider SP's, they can reduce the costs associated with their ISP's using this hierarchical Peer-to-Peer infrastructure since they reduce their traffic in the links contracted with their ISP's. Furthermore, this approach can be also used for community networks that cannot place any central entity in any place and H-P2PSIP allows having a distributed infrastructure to allow conferencing among users of the same community and also with users of other communities or domains.

6.9.2 Adaptation of Peer-to-Peer overlay networks to mobility

Mobility in nodes is a behaviour that is becoming more popular in the current Internet. Its impact is really relevant in the current communication protocols. If the consequences of mobility are carefully considered, it can be deduced that they have a great impact on Peer-to-Peer overlay networks since mobility increases the churn in any overlay network and consequently affects to their performance. Therefore, a brief summary about mobility is provided in the next paragraphs in order to see how mobility can affect to overlay networks and later a solution based on H-P2PSIP is given as possible mechanism for adaptation to different mobile environments. This solution is published in [MYBG09].

6.9.2.1 Mobility

Mobility is a characteristic that is being more usual in user terminals and devices. This feature allows connectivity wherever and whenever some access technology is available for it. This feature can be summarised with the famous concept *always-on*. Although the provision of mobility is common nowadays (GSM, 3G, Mobile WiMax, ...) it is not a trivial

functionality. Many studies have been performed to achieve seamless mobility; however, this fact is impossible to obtain completely. It is usual to have some disruption in the communication of a terminal when is changing from one cell to another or from a technology to another.

A first classification of mobility support solutions that can be considered is the existence of macro-mobility and micro-mobility. Many definitions can be used to explain these terms but, in a simple way, we consider them as follows. Macro-mobility is the mobility of terminals between different domains. The concept of domain here is quite wide. In this context, we mean a part of the network where mobility can be managed with a local solution, a micro-mobility solution.

Micro-mobility happens when the movement is performed inside a domain (i.e. adjacent cells of the same network). Thus, micro-mobility manages mobility closer to the terminal and implies a faster resolution of the connectivity disruption in the terminals. Several proposals have been studied to solve this problem [ASB+06], [CGC00], [SCMB05]. These types of mobility are usually associated to the access technology used by the terminals. For instance, in UMTS, micro-mobility implies the management of changing from one cell to another, whereas macro-mobility implies the movement of a terminal from one operator network to another or changing from one access technology to another. Obviously, the time needed for macro-mobility handovers is larger with respect to micro-mobility handovers.

6.9.2.2 Mobile IP

Mobility in IP networks implies the need to change the IP address of the moving terminal each time it moves to a new network. Micromobility solutions can hide or avoid this change of the IP address if the movement is within a micromobility domain. Nevertheless in other cases, i.e. without a micromobility solution or when changing the micromobility domain, the terminal needs to change the IP address when moving. The reason is that IP addresses act as locators of the terminal and must have a value according to where the terminal is connected to the network.

An additional problem is that IP addresses are not only locators, they also act as identifiers. This means that to keep ongoing communications, a moving terminal requires a permanent IP address as part of the identifier of its communications. The IETF has standardised solutions to support IP mobility both for IPv4 [Per02] and IPv6 [JPA04] that work by associating with the terminal a permanent address that acts as identifier (the Home Address, HoA), and temporal addresses that act as locators and that the terminal configures in the visited networks (Care of Addresses, CoA). A new entity, the Home Agent (HA) is introduced to act as rendezvous point for the communications of the terminal using the HoA. The HA is situated where the HoA is topologically valid and forwards packets to the mobile terminal. Furthermore, in order to accelerate the signalling with the HA, a strategy based on anchor points can be adopted [SCMB05]. It must be considered that these optimisations must be done per each flow that it was established before the movement.

6.9.2.3 Peer-to-Peer Overlays and Mobility

Once that the topic of mobility has been shortly reviewed in the previous section, we can consider how the mobility affects to DHT overlay networks.

The mobility affects to the performance of Peer-to-Peer networks for two reasons. First of all, we have the service disruption because of handovers. Depending on the type of handover, macro-mobility or micro-mobility based, this time will be different and will affect in a major or minor way to the performance of the overlay network. Although mobility protocols try to minimise this effect, typically we will always have a certain level of impact of the handovers in the performance. Furthermore, we have to take into account another fact: depending on the mobility solution a change of IP address can be needed when the terminal moves. For example if a terminal uses Mobile IP but it wants to register the CoA instead of the HoA in the DHT Peer-to-Peer network to avoid routing inefficiencies of using the HoA. Therefore, a modification in the maintenance algorithm of the DHT needs to be considered. This implies that the overlay routing tables have to be updated more frequently, and the maintenance traffic needed to update these overlay routing tables will also increase. If in addition to this problem, we consider that mobile nodes usually have limited bandwidth capabilities, the increment in the maintenance traffic does not seem to be a good solution. Furthermore, the mobile IP handovers also introduce disruptions in the connectivity, these disruptions increment the churn suffered by the Peer-to-Peer overlay network. Therefore, it would be desirable to minimize these effects as much as possible.

6.9.2.4 Management of routing tables in Peer-to-Peer overlay networks

Considering section 2.4.3 in the State of Art chapter, basically two strategies apply to manage routing tables: proactive and reactive. The first approach is interesting for scenarios with high churn because the traffic generated to update the routing tables is limited by the periodicity that is used to refresh the entries. On the other hand, the second approach is suitable for scenarios where the churn is low. Only maintenance traffic is generated if necessary, and the errors caused are minimal because they do not occur frequently.

Finally, when some peers have a very high churn, it is better that they do not participate in the maintenance of the overlay network. Their churn will produce more drawbacks than the benefits of their resources to the overlay network. The solution is to allow these peers to use the overlay, but not to participate in its maintenance [MBRM06], these peers are called clients in P2PSIP [JLR+09a].

6.9.2.5 Management of Peer-to-Peer routing tables in mobile environments

The question that is discussed in this section is which is the most suitable strategy that must adopt a Peer-to-Peer overlay network if it is not desired to reduce the performance in a heterogeneous scenario with mobile peers. Several considerations can be done. One could consider using the approach of using the client profile for mobile nodes, so these peers would not participate in the overlay network [MBRM06]. However, this approach

cannot be applied in a scenario where only mobile peers exist. In this case, it would be more suitable a proactive strategy in order to minimise the maintenance traffic of updating the overlay routing entries and avoiding as much as possible of the wireless interfaces of the peers. Nevertheless, in a heterogeneous scenario, stable peers will have to increase the costs of their maintenance traffic since mobile nodes exist, although a reactive strategy would be more suitable. Therefore, depending on the scenario one approach would be more suitable than the other one. Furthermore, we cannot predict how new services will evolve and which strategy would be the best, the complexity of this problem is evident. Thus, we advocate for a flexible solution that can be adopted in any scenario. A classification of the different nodes participating in a Peer-to-Peer network can be done; this classification allows planning the overlay structure and the associated parameters according to the profile of each group. One classification according to their mobility can be as follows:

- **Fixed Nodes**
  - *Stable Nodes*: These nodes present large up-times and a stable connectivity. This fact usually implies a fixed available bandwidth and RTT in the access network.
  - *Unstable Nodes*: These nodes present small up-times. This behaviour is usually because of connectivity problems or own system instability. Bandwidth and RTT are usually stable but only available in short periods of time.
- **Mobile Nodes**
  - *Low Mobility Nodes*: This profile considers those nodes that have mobility support but they don not change their location very frequently. Although the bandwidth and RTT are given by the access network, they depend on the number of users that are connected in a cell or access point.
  - *High Mobility Nodes*: These nodes usually change their cell or visiting network since they change their location really fast. This pattern implies many disruptions. Therefore, the RTT and bandwidth are heterogeneous and difficult to predict because of the continuous changes.

A different Peer-to-Peer overlay network can be built according to the different groups listed before and the most suitable strategy and DHT overlay network [LSM+05] can be used. For fixed nodes, we can use a reactive strategy, but for Unstable and Low Mobility Nodes, both profiles with a higher churn, we can use a reactive algorithm tweaked to each one of these profiles. Finally, high mobility peers can be configured as clients that are attached to the overlays maintained by the other profiles if necessary. Thus, the problem that arises is how to allow the communication among the different overlay networks. This problem can be solved with H-P2PSIP if we do an intelligent mapping of the different profiles in the H-P2PSIP architecture. Furthermore, this solution gives a great flexibility than can be really interesting for future deployments. The main drawback than can be related with this solution is the fact that probably is not a very good idea to have only mobility peers in an overlay network because their lifetimes probably would be short and the stability of super-peers peers could be affected in a dramatic way. This last statement depends on

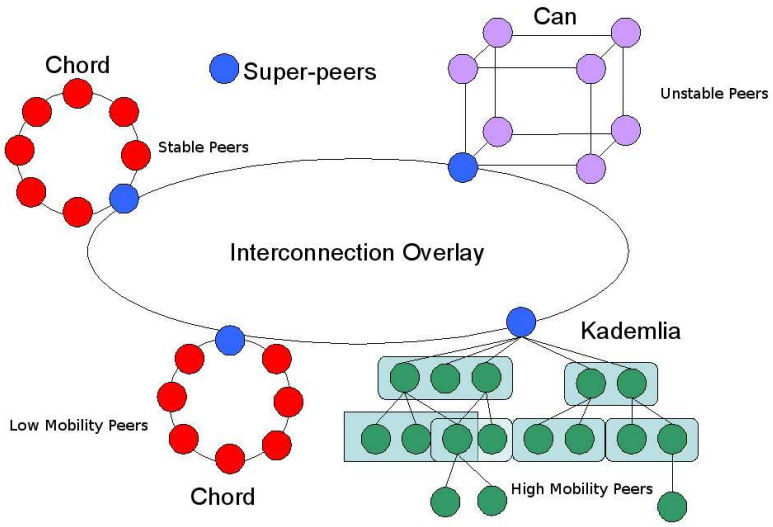


Figure 6.4: *H-P2PSIP providing Mobility Enhancement*

the strategies adopted for that profile. However, more stable peers can be introduced but they will find drawbacks because they are not attached on their original overlay network. Therefore, some type of incentive mechanism should be applied. Incentives in Peer-to-Peer systems is an open topic and it is out of scope with respect to this Thesis and therefore it is not analysed but it should be probably considered in a production implementation.

6.9.2.6 H-P2PSIP in mobile environments

The solution we propose is to change how resources are stored in H-P2PSIP in order to maintain different overlay networks associated to the same domain where each overlay network is composed by a different profile of peers. Using this approach, each Peer-to-Peer network can be optimised according to the specific profile of the peers; an example is given in Figure 6.4. The peers in the same overlay share the same mobility profile and the connectivity between peers with different profiles is allowed through the Interconnection Overlay.

In order to take into account the mobility profile of peers, a mobility tag can be included in the URI's to identify which mobility profile would be used with that URI. The defined format is as follows: *user@example.org:**xx***. The *xx* tag defines where a user is attached and this tag can be *st* (**st**able peer), *un* (**un**stable peer), *lm* (**l**ow **m**obility peer) and *hm* (**h**igh **m**obility peer).

URI's are mapped to Hierarchical-ID's in the following manner. The Prefix-ID is obtained by applying a hash to the domain of the URI and the profile tag: `Prefix-ID=hash(example.com:xx)`. The Suffix-ID is obtained from the hash of the URI without the profile tag: `Suffix-ID=hash_a(resource@example.com)`.

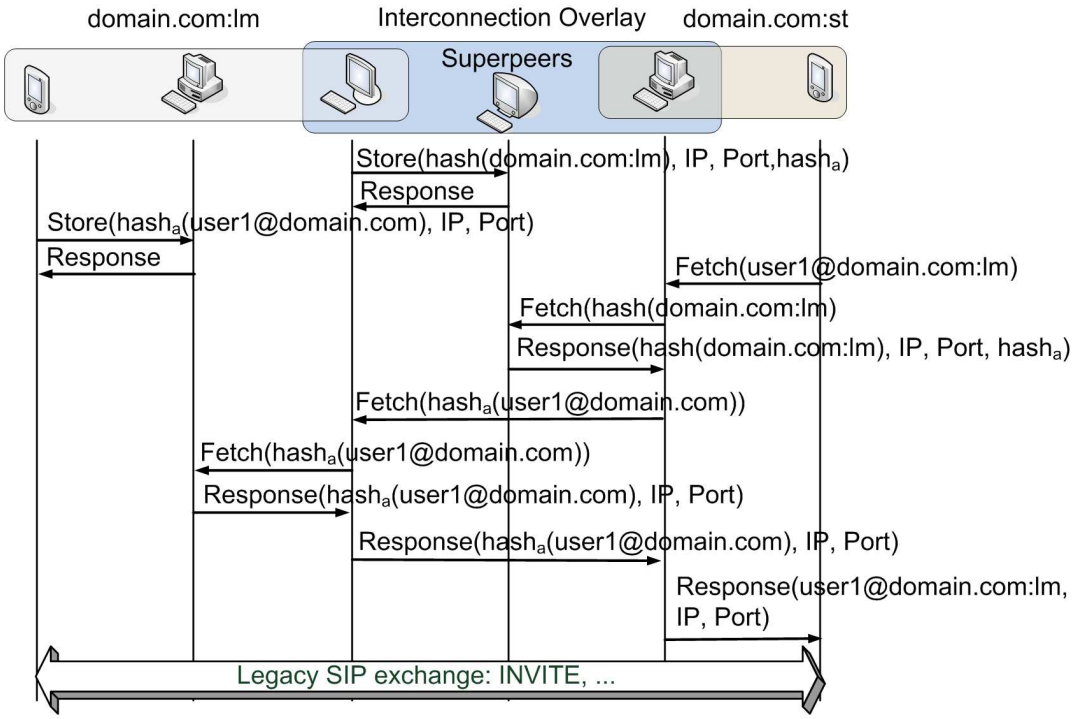


Figure 6.5: *H-P2PSIP signalling in mobile environments*

When storing an object in the overlay network, each Resource-ID has a Hierarchical-ID associated with the original URI and the resource information.

The signalling is detailed in Figure 6.5. The differences with respect the example in Figure 6.3 are the usage of the mobile tags in the URI's and the fact the signalling in Figure 6.5 is among to different overlay networks but both of them belong to the same domain. No additional details are provided in relation with this signalling exchange since this one is similar as the signalling detailed in section 6.7. Only the mapping function to generate the Hierarchical-ID is modified, therefore the signalling does not need to be modified since it makes uses only of Hierarchical-ID's. Finally, if a resource has not been found in a mobility profile, the other different profiles can be queried through the super-peer that is attached in the Interconnection Overlay. In order to accelerate the speed of the search, the query can be sent to all the overlay networks supporting the different mobility profiles.

6.9.2.7 Dynamic profile update

A problem that arises with this proposal is how to contact with a peer with an unknown mobility profile. In order to avoid losing time and bandwidth with unnecessary queries, a peer can leave the information of its new position in the last visited domain. This information will be only available for a certain period of time. This solution is a compromise between looking for peers among all the domains and to store the location information in each one of

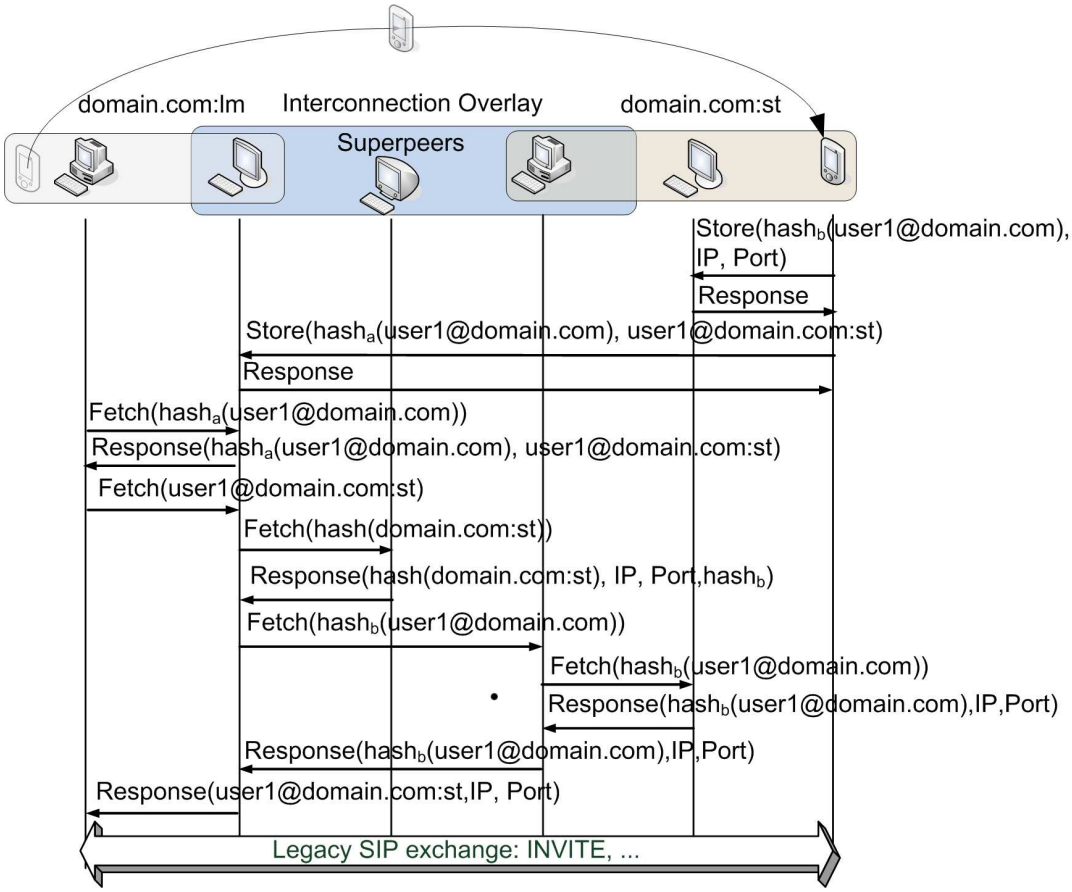


Figure 6.6: *Dynamic Update Signalling*

the domains.

The way to proceed is as follows and it is illustrated on Figure 6.6. If a peer changes its location from the domain of low mobility to the domain of stable peers, it has to register this information in the new domain. Additionally, it has to register in the previous overlay domain a pointer to its new attachment point. In Figure 6.6, its URI with its new profile tag is stored on the original domain. If a peer looks for it in the old domain, it obtains the pointer of its new location. Thus, it can start the same signalling exchange as explained in Figure 6.5 to obtain its contact information. Once these actions have been performed, a legacy SIP exchange can be done between the partners of the new session.

6.10 Conclusions

This chapter contains a hierarchical architecture to support a hierarchical DHT; it is named H-P2PSIP. This design is based on P2PSIP that allows the implementation of any DHT. The hierarchy is based on a Hierarchical-ID composed by a Prefix-ID and a Suffix-

ID. The Suffix-ID is used to route the information inside each Peer-to-Peer overlay network whereas the Prefix-ID is used for routing in the Interconnection Overlay. If a peer wants to look for an item whose Prefix-ID is different to its own Prefix-ID, it has to forward a query to its super-peer. The super-peers in H-P2PSIP are attached to their own overlay (lower level of the hierarchy) and also to the Interconnection Overlay (upper layer of the hierarchy). The Interconnection Overlay plays the role of directory and stores what super-peers are taking care of each domain. Therefore, any super-peer can find the super-peers associated with a Prefix-ID and each super-peer can forward queries to another super-peer, which is attached in the destination overlay network. In order to facilitate the communication with different overlay networks, the RELOAD protocol defined by the P2PSIP WG is used since it offers a common payload and format to define the parameters of different overlay networks, queries and other desired functionalities that in other case would be very difficult to unify. The necessary signalling for the hierarchical architecture is defined taking into account the routing based on the Hierarchical-ID. This design is published in [MYBG+08a], [MYBC+09]. Finally, some usage scenarios are detailed [MYBG09].

Amistades que son ciertas nadie las puede turbar.

Miguel de Cervantes (1547-1616)

Chapter 7

Analytical Evaluation of H-P2PSIP

This chapter provides an analytical study of the H-P2PSIP proposal presented in the previous chapter.

7.1 Introduction

The work presented in this chapter consists in validate analytically the proposal presented in the previous chapter. It demonstrates that the proposed H-P2PSIP architecture is a good solution from the point of view of the Routing Performance and Routing State. Although the main feature of the design is to allow the exchange of information among different overlays, this exchange should be as efficient as possible in order to avoid the unnecessary consumptions of bandwidth and resources. This analytical evaluation has been published in different conferences and journals [MYCGM08], [MYGCM09], [MYBG+08b], [MYBG+08a], [MYBC+09].

7.2 Routing Performance in H-P2PSIP

This section studies the Routing Performance in a system based on H-P2PSIP but first of all it is necessary to define the parameters that are considered in the analytical model. In the next list, there is a definition of these parameters:

- K: The number of P2PSIP domains.
- M_k: The number of peers in P2PSIP domain k.
- N: Peers from all the P2PSIP domains. In our case, it is considered that a peer cannot be attached to multiple P2PSIP domains, hence $N = \sum_{i=1}^{K} M_i$.

- S_k: The number of super-peers in P2PSIP domain k.
- ρ_{ij}: The probability of launching a query from the P2PSIP domain i to the P2PSIP domain j.
- $C(x)$: The number of hops needed to find a super-peer in the Interconnection Overlay depending on the number of super-peers *x*. This value depends on the type of overlay used in the Interconnection Overlay.
- $D_k(x)$: The number of hops needed to find a peer in overlay network k as function of the number of peers *x* belonging to the P2PSIP domain.

We assume that all the peers in a P2PSIP domain know their super-peers from the Interconnection Overlay. This assumption implies that only *one hop* is needed to reach the super-peer. The cost in number of hops of an intra-domain query (restricted to the own domain) is bounded by the overlay network used in the domain. On the other hand, an inter-domain query (among different domains) must be routed to other domain. Therefore, it would be in any case one hop to reach any of the super-peers plus the hops needed in the Interconnection Overlay plus the hops needed in the destination domain. Therefore, taking into account the previous considerations, we can calculate the Routing Performance (RP) of this DHT-based hierarchical overlay networks. First of all, we define the cost of finding a peer in each overlay network:

- $D_k\left(M_k\right)$: The cost of finding a peer in a P2PSIP domain.
- $C\left(\sum_{k=1}^{K} S_k\right)$: The cost of finding a super-peer in the Interconnection Overlay.

If the probability of obtaining an item in a domain from its super-peer is considered negligible since the average number of peers in a P2PSIP domain is very large in comparison with the number of super peers ($M_k \gg S_k$), the average Routing Performance obtained by a peer in P2PSIP domain i can be written as follows:

$$RP_i = \rho_{ii} \cdot D_i\left(M_i\right) + \sum_{j=1, j\neq i}^{K} \rho_{ij} \cdot \left[1 + D_j\left(M_j\right) + C\left(\sum_{k=1}^{K} S_k\right)\right] \tag{7.1}$$

The first term of the sum is the cost of searching something in the P2PSIP domain of a peer, whereas the second term is the cost for the searches in the other P2PSIP domains. Each term is bounded by the probability of occurrence of each case.

The average number of hops among all the domains participating in the hierarchical overlay is given by the next expression:

$$RP = \frac{1}{N} \cdot \sum_{i=1}^{K} M_i \cdot RP_i \tag{7.2}$$

Finally, if the number of peers is the same in all P2PSIP domains ($M_k = M$), we have:

$$RP = \frac{1}{K} \cdot \sum_{i=1}^{K} \cdot RP_i \tag{7.3}$$

7.2.1 Random Independent Queries

If we assume that the number of peers is equal in all P2PSIP domains and each look-up in the overlay is considered randomly independent, we obtain that the probability of looking for a peer attached to other P2PSIP domain is equally distributed among all the foreign P2PSIP domains. This means that $\rho_{ii} = \rho_{ij} = \frac{1}{K}$ and we obtain Equation 7.4 from Equation 7.1:

$$\begin{aligned} RP_i = & \frac{1}{K} \cdot D_i\left(M\right) + \\ & + \sum_{j=1, j\neq i}^{K} \cdot \frac{1}{K} \cdot \left[1 + D_j\left(M\right) + C\left(\sum_{k=1}^{K} S_k\right)\right] \end{aligned} \tag{7.4}$$

Finally, if the same overlay is used in all P2PSIP domains the sum can be eliminated from Equation 7.4 and RP_i becomes equal to RP:

$$\begin{aligned} RP_i = RP = & \frac{1}{K} \cdot D\left(M\right) + \\ & + \frac{K-1}{K} \cdot \left[1 + D\left(M\right) + C\left(\sum_{k=1}^{K} S_k\right)\right] = \\ & = D\left(M\right) + \frac{K-1}{K} \cdot \left[1 + C\left(\sum_{k=1}^{K} S_k\right)\right] \end{aligned} \tag{7.5}$$

7.2.2 Intra-domain queries more likely than Inter-domain queries

However, the probability of looking for a peer in the own domain can be different from the one of looking for a peer in other P2PSIP domains. Thus, the inter-domain query probability is $\rho_{ij} = \frac{1-\rho_{ii}}{K-1}$ and we can express Equation 7.1 as follows:

$$\begin{aligned} RP_i = & \rho_{ii} \cdot D_i\left(M\right) + \\ & + \sum_{j=1, j\neq i}^{K} \cdot \frac{1-\rho_{ii}}{K-1} \cdot \left[1 + D_j\left(M\right) + C\left(\sum_{k=1}^{K} S_k\right)\right] \end{aligned} \tag{7.6}$$

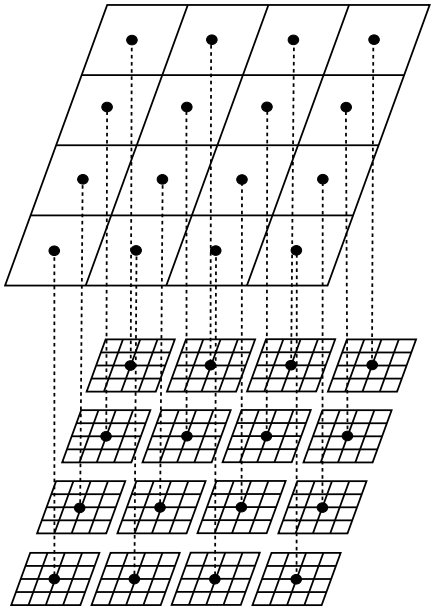


Figure 7.1: *Hierarchical CAN overlay network*

This expression is useful for some type of scenarios like VoIP in community networks where $\rho_{ii} > \rho_{ij}$, which implies that calls between peers that belong to the same company or social network are more likely.

If the same overlay is used on all the P2PSIP domains the sum can be eliminated from Equation 7.6 and RP_i becomes equal to RP:

$$\begin{aligned} RP_i = RP = \rho_{ii} \cdot D(M) + \\ + (1 - \rho_{ii}) \cdot \left[1 + D(M) + C\left(\sum_{k=1}^{K} S_k \right) \right] = \\ = D(M) + (1 - \rho_{ii}) \cdot \left[1 + C\left(\sum_{k=1}^{K} S_k \right) \right] \end{aligned} \tag{7.7}$$

We define ρ_{ii} as the intra-domain hit probability and it defines the probability of establishing a connection inside the own domain.

7.3 H-P2PSIP in CAN

In this section, we study the performance of a hierarchical CAN overlay with two levels as it is shown on Figure 7.1. If $C(x) = d_i x^{\frac{1}{d_i}}$ and $D(x) = d_l x^{\frac{1}{d_l}}$ are used on Equation 7.5

with $S_k = 1$, we have the Routing Performance:

$$RP = d_l \left(\frac{N}{K}\right)^{\frac{1}{d_l}} + \frac{K-1}{K}\left(1 + d_i K^{\frac{1}{d_i}}\right) \tag{7.8}$$

Equation 7.8 presents the Routing Performance for a hierarchical CAN overlay where d_l is the number of dimensions for the domains and d_i is the number of dimensions for the Interconnection Overlay. Each value can be optimised independently considering the number of peers in each level. Thus, it can be written that $d_l = \ln \frac{N}{K}$ and $d_i = \ln K$ (these value can be obtained minimising the Routing Performance of the canonical CAN overlay network). However, if the development of a hierarchical CAN based application is considered, it would be reasonable to have the same value of d in both levels of the hierarchy because just one version of CAN needs to be developed and the code can be easily reused. Taking this fact into account, Equation 7.9 presents the Routing Performance when only one value of d is considered:

$$RP = d \left(\frac{N}{K}\right)^{\frac{1}{d}} + \frac{K-1}{K}\left(1 + dK^{\frac{1}{d}}\right) \tag{7.9}$$

Thus, the optimum configuration parameters for the hierarchical CAN overlay network and some general design rules have been obtained. Two parameters can be modified in order to get a satisfactory system performance: the number of domains K and the number of dimensions d of the overlay network.

Theorem 1. *If $K \gg 1$, then the optimum value of K is $K \simeq \sqrt{N}$.*

Proof. If $K \gg 1$, Equation 7.9 can be rewritten as:

$$RP = f(K) = d \left(\frac{N}{K}\right)^{\frac{1}{d}} + 1 + dK^{\frac{1}{d}} \tag{7.10}$$

In order to obtain the best configuration parameter for K, the first derivate respect to K is:

$$f'(K) = -\frac{1}{K}\left(\frac{N}{K}\right)^{\frac{1}{d}} + K^{\frac{1}{d}-1} \tag{7.11}$$

$f'(K)$ is equal to 0 when $K = \sqrt{N}$. This point will be a minimum if the second derivate is positive on this point. Thus, the second derivate is obtained:

$$f''(K) = \left(1 + \frac{1}{d}\right)\frac{1}{K^2}\left(\frac{N}{K}\right)^{\frac{1}{d}} + \left(\frac{1}{d} - 1\right)K^{\frac{1}{d}-2} \tag{7.12}$$

If $f'(K) = 0$ is a minimum, it implies that $f''(K) > 0$. Therefore, it can be written in the following inequalities:

$$\begin{aligned} \left(1 + \frac{1}{d}\right) \frac{1}{K^2} \left(\frac{N}{K}\right)^{\frac{1}{d}} &> \left(1 - \frac{1}{d}\right) K^{\frac{1}{d}-2} \\ \frac{N}{K^2} &> \left(\frac{d-1}{d+1}\right)^d \end{aligned} \tag{7.13}$$

Considering that $f'(K) = 0$ when $K \simeq \sqrt{N}$, if this value of K is substituted on Equation 7.13, it is obtained that:

$$1 > \left(\frac{d-1}{d+1}\right)^d \tag{7.14}$$

So, Equation 7.14 is valid for $\forall d \in (1, \infty)$ and it can be said that $K = \sqrt{N}$ is a minimum. □

This result is important not only because it gives a numerical value for the optimal number of domains K in the hierarchical CAN overlay network, also due to the fact that this value is *independent* of the number of dimensions d.

Theorem 2. *Only one value of d minimises the number of hops on a hierarchical CAN overlay network with N peers and K domains.*

Proof. The first derivate of Equation 7.9 with respect to d is calculated to study if a minimum exists:

$$\begin{aligned} f'(d) = \left(\frac{N}{K}\right)^{\frac{1}{d}} \left(1 - \ln\left(\frac{N}{K}\right)^{\frac{1}{d}}\right) + \\ + \frac{K-1}{K} K^{\frac{1}{d}} \left(1 - \ln K^{\frac{1}{d}}\right) \end{aligned} \tag{7.15}$$

If we take into account that $d \in (1, \infty)$, then:

$$f'(d=1) = \frac{N}{K}\left(1 - \ln\frac{N}{K}\right) + (K-1)(1 - \ln K) \tag{7.16}$$

and

$$\lim_{d\to\infty} f'(d) = 1 + \frac{K-1}{K} \tag{7.17}$$

Thus, if $\frac{N}{K}, K > e$ (this fact occurs always, K=2 has no sense for our scenario), then $f'(d=1) < 0$ and $f'(d=\infty) > 0$ which implies that one or more solutions exist for $f'(d) = 0$ in our range of work.

The next step is to prove the unicity of the solution. Then, the second derivate respect to d is calculated:

$$f''(d) = \frac{1}{d}\left(\frac{N}{K}\right)^{\frac{1}{d}}\left(\ln\left(\frac{N}{K}\right)^{\frac{1}{d}}\right)^2 + \frac{K-1}{K}\frac{K^{\frac{1}{d}}}{d}\left(\ln K^{\frac{1}{d}}\right)^2 \tag{7.18}$$

Taking into account again that $f''(d) > 0, \forall d \in (1,\infty)$, then $f'(d)$ is a monotonous increasing function and there is only one solution of $f'(d) = 0$ and this value is a minimum. □

Theorem 3. *If the optimum value of K is used ($K \simeq \sqrt{N}$), then $d = \ln\frac{N}{K} = \ln K$ is the optimum value of d parameter.*

Proof. The following variable substitution can be done:

$$A = \left(\frac{N}{K}\right)^{\frac{1}{d}} \tag{7.19}$$

$$B = K^{\frac{1}{d}} \tag{7.20}$$

The above substitution when applied to Equation 7.15 results with:

$$f'(d) = A(1 - \ln A) + B(1 - \ln B) \tag{7.21}$$

Furthermore, if A and B are transformed to find a relation between them:

$$\frac{1}{d} = \frac{\ln A}{\ln\frac{N}{K}} \tag{7.22}$$

$$\frac{1}{d} = \frac{\ln B}{\ln K} \tag{7.23}$$

Equalising the last two equations we have:

$$\ln B = \frac{\ln K}{\ln\frac{N}{K}}\ln A \tag{7.24}$$

If a variable x is defined as $x = \dfrac{\ln K}{\ln \dfrac{N}{K}}$, it is obtained that $B = A^x$. With this result, we can rewrite Equation 7.21 as:

$$f'(d) = A(1 - \ln A) + \frac{K-1}{K} A^x (1 - \ln A^x) \tag{7.25}$$

$f'(d) = 0$ cannot be solved analytically but can be solved using the bisection method. Nevertheless, if is considered that the optimum value for K is $K = \sqrt{N}$, then $\dfrac{N}{K} = K$ and $x = 1$. Thus, Equation 7.25 can be expressed as:

$$f'(d) = \left(1 + \frac{K-1}{K}\right) A(1 - \ln A) \tag{7.26}$$

For $f'(d) = 0$, it is obtained that $A = e$. If this value is used on Equation 7.22, then we have that $d = \ln\left(\dfrac{N}{K}\right) = \ln(K)$ and *Theorem 3* is proved. □

Corollary 1. *The best Routing Performance that can be found for a hierarchical CAN overlay is:*

$$RP = \frac{\sqrt{N}-1}{\sqrt{N}} + \left(1 + \frac{\sqrt{N}-1}{\sqrt{N}}\right) \ln \sqrt{N} (\sqrt{N})^{\frac{1}{\ln \sqrt{N}}}$$

Proof. The best RP can be obtained if Theorem 1 (optimum value of K) and Theorem 3 (optimum value of d) are applied to Equation 7.9. □

Corollary 2. *If a hierarchical CAN overlay is configured with its optimum parameters, the number of dimensions d is a half of the number of dimensions for the optimum flat counterpart.*

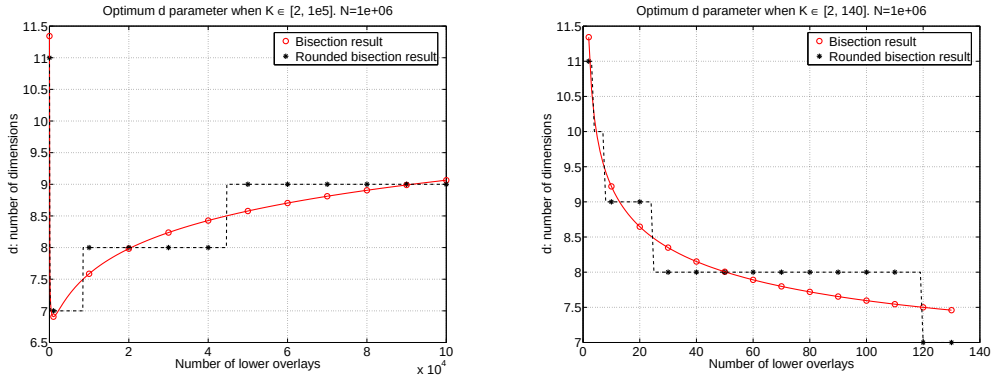
Proof. The proof is obtained by mathematical manipulation of the result given on Theorem 3:

$$d_{hierarchical} = \ln K = \ln \sqrt{N} = \frac{1}{2} \ln N = \frac{1}{2} d_{flat} \tag{7.27}$$

□

Corollary 2 means that only a half of the Routing State of the canonical CAN overlay network is needed for its hierarchical counterpart.

Corollary 3. *If $\sqrt{N} \gg 1$, the optimum Routing Performance of a hierarchical CAN overlay network is one hop greater than the optimum Routing Performance of a flat CAN overlay network.*



(a) Optimum value of d for a wide range of K

(b) Optimum value of d for small values of K

Figure 7.2: *Optimum number of dimensions depending on the number of domains*

Proof. If we take the best Routing Performance that can be achieved to a hierarchical CAN overlay network from Corollary 1 and given that $\sqrt{N} \gg 1$:

$$\begin{aligned} RP &= 1 + 2\ln\sqrt{N}(\sqrt{N})^{\frac{1}{\ln\sqrt{N}}} = 1 + \ln N(N)^{\frac{1}{\ln N}} = \\ &= 1 + d_{flat}N^{\frac{1}{d_{flat}}} = 1 + RP_{flat} \end{aligned} \tag{7.28}$$

□

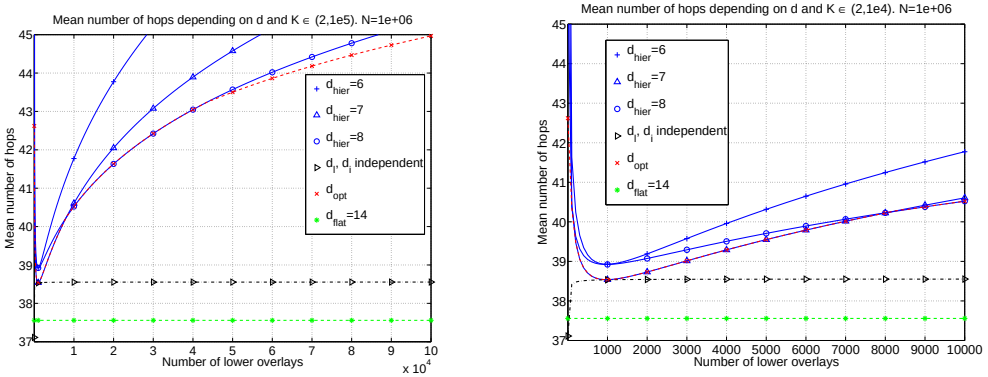
Thus, the Routing Performance of a hierarchical CAN overlay network cannot be better than the flat Routing Performance according to Corollary 3, although the difference is small.

7.3.1 Hierarchical CAN Design Rules

Theorem 3 can be used when a distributed application based on a hierarchical overlay network is able to change the number of domains dynamically. For instance, in a file-sharing network the number of predefined domains could be configured dynamically according to the number of peers attached to the network, if some mechanism is provided for this purpose.

On the other hand, Equation 7.25 is useful when the number of domains cannot be configured dynamically and it is fixed in advance. One example of this scenario could be the P2PSIP scenario. In this scenario, the number of groups is given by the number of domains subscribed to the Interconnection Overlay. Thus, it would be interesting to know the range of work around the optimum configuration value of the d parameter.

In order to understand the importance of the number of dimensions d on a hierarchical CAN overlay network, Equation 7.25 has been solved using the numerical method of the bisection. The value of $N = 10^6$ has been used for this analysis and a large range of K



(a) Routing Performance for a wide range of K (b) Routing Performance zoom around the minimum

Figure 7.3: *Routing Performance for hierarchical CAN overlay*

values has been explored. The results are shown on Figure 7.2(a), the red solid line is the result of the bisection method and the black dashed line is the nearest integer to the bisection results for each value of K. It can be observed how the value of d changes depending on K. The value of $d = 7$, which is the value for the best performance point according to Theorem 3, is reached quickly as it can be appreciated on Figure 7.2(b). This value of d is valid for $K \in \left(120, 10^4\right)$. This implies a wide range of work along K for the optimum value of d.

Furthermore, in order to see the performance of a hierarchical CAN overlay, the Routing Performance complexity for an optimum flat CAN overlay is plotted in Figure 7.3 with a green dashed line with asterisks ($d = 14$). The Routing Performance for a hierarchical CAN overlay network with the optimum values of d_l and d_i is in a black dashdot line with triangles and the Routing Performance of a hierarchical CAN overlay is also plotted when the same value of d is used on both levels of the hierarchy with blue solid lines. Several markers have used for the different values of d as it is indicated on the legend.

In addition to this information, the Routing Performance for the best configuration for the d parameter according to Equation 7.25 has been plotted on a red dashed line with crosses in Figure 7.2(a). The x axis represents the number of K domains and the legend of each figure explains what values of d have been used for each line.

The values used to generate Figure 7.3(a) are $N = 10^6$ nodes and $K \in [2, 10^5]$. In this figure, the behaviour of the Routing Performance is shown along a wide range of K. On one hand, the figure shows how an optimum flat CAN overlay $(d = 14)$ has a better performance than its hierarchical counterpart but it needs twice the number of dimensions of a hierarchical CAN overlay ($d = 7$). Nevertheless, the difference for this optimum value is negligible in comparison with the flat CAN overlay counterpart, only one hop larger.

The minimum predicted by Theorem 3 can be seen clearly on Figure 7.3(b). An important fact is that the red dashed line and the black dashdot line meet on this point. The black dashdot line is the Routing Performance when each overlay is configured with the

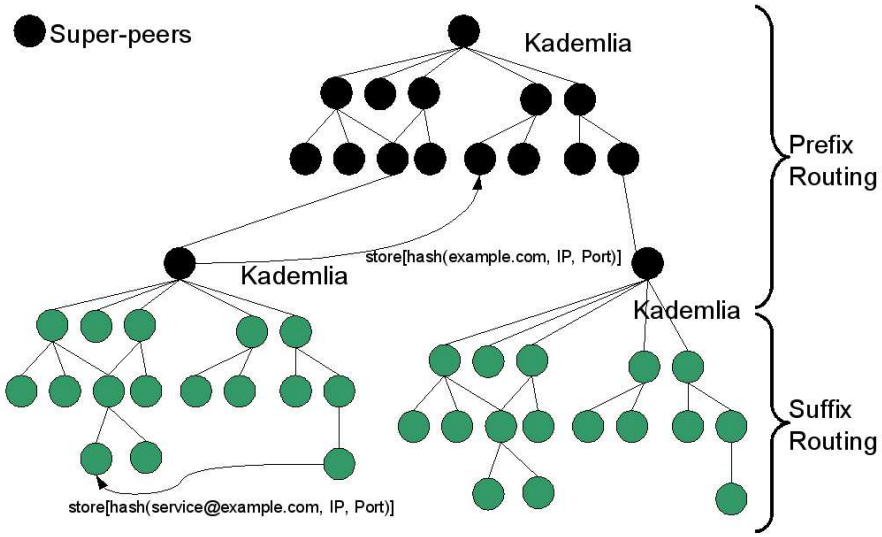


Figure 7.4: *Hierarchical Kademlia Overlay Network*

optimum value of d independently. This fact means that around the optimum of d we have $d = d_i = d_l$. Thus, $d \simeq \ln \frac{N}{K} \simeq \ln K$ and $N \simeq K^2$ as it has been previously predicted on Theorem 3. This value is around $K = 1000$ as expected. Moreover, it can be observed that for $K \in [100, 10000]$, the Routing Performance only decreases two hops with respect to the optimum value. Thus, the optimum configuration parameter of d has a good range of work along K and the decrement of performance is negligible. In other case, the Routing Performance should be checked in Figure 7.3(a).

7.4 H-P2PSIP in Kademlia

In this section, we study the H-P2PSIP Routing Performance and Routing State in the case when a Kademlia overlay [MM02] is used in all the P2PSIP domains and also on the Interconnection Overlay (see Figure 7.4). Kademlia has been selected, because it is one of the most used DHT overlays in p2p applications like eMule, Bittorrent, etc.

Summarising, Kademlia is an overlay network, which has a Routing Performance and a Routing State with a logarithmic dependency on the number of peers from the overlay network. These results are due to its XOR distance-based routing algorithm.

In order to verify the efficiency of our solution, when the Kademlia protocol is used, we use the next equality: $C(x) = D(x) \sim \log_B x + c$. We substitute this expression in Equation 7.5 because the validation is performed via simulation with a setup similar to the conditions

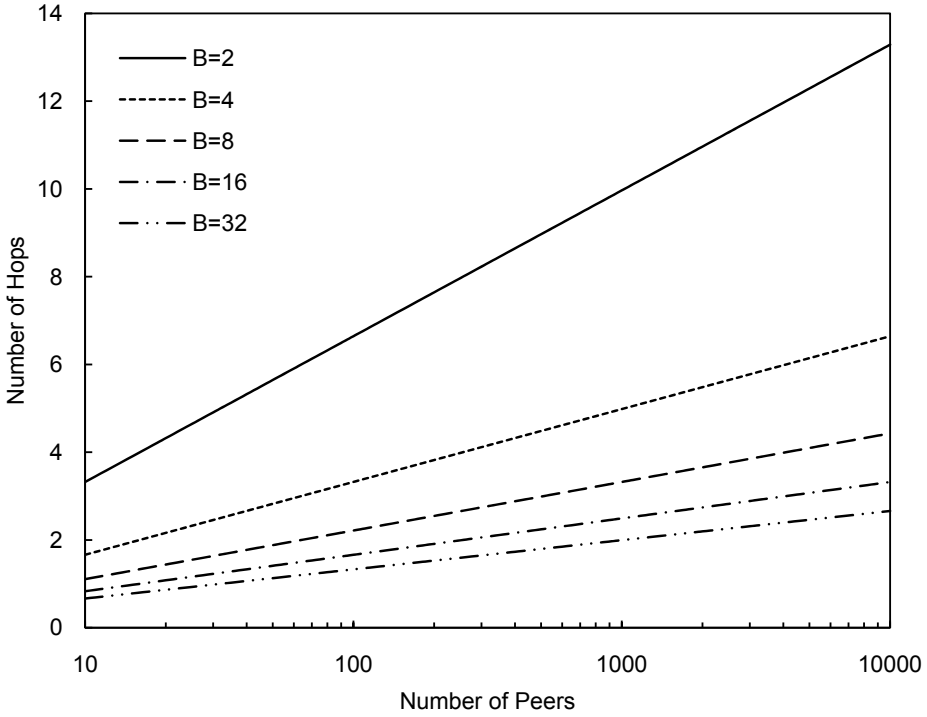


Figure 7.5: *Routing Performance*

which are valid for this expression. Therefore:

$$RP = RP_i \sim \log_B (M) + c + \frac{K-1}{K} \cdot \left[1 + \log_B \left(\sum_{k=1}^{K} S_k \right) + c \right] \tag{7.29}$$

If $K \gg 1$ and taking into account the properties of the logarithm, we can write:

$$RP = RP_i \sim \log_B (M) + c + 1 + \log_B \left(\sum_{k=1}^{K} S_k \right) + c = 1 + \log_B \left(M \cdot \sum_{k=1}^{K} S_k \right) + 2c \tag{7.30}$$

In Figure 7.5 we can see the Routing State taking into account up to 10^4 peers. The x-axis is the number of peers and the y-axis represents the number of hops. To determine the Routing Performance of a Kademlia-based P2PSIP domain, we have to see how many peers belong to the overlay in order to see the required number of hops. The same method can be applied to the Interconnection Overlay if we consider $S_k = 1$. Furthermore, the total number of hops for the overlay can be estimated considering all the peers in all the P2PSIP domains.

Since the Routing State must also be taken into account, the number of entries depends on the number of peers and on the setup parameter B. The number of routing in

a super-peer is $O\left(\log_B\left(M \cdot \sum_{k=1}^{K} S_k\right)\right)$. If a flat overlay is used to connect all peers in different P2PSIP domains, peers would need $O\left(\log_B\left(M \cdot K\right)\right)$ routing entries, but using the hierarchical architecture, peers only need $O(\log_B M)$. Thus, legacy peers save $log_B(K)/log_B(M \cdot K) \cdot 100\ \%$ of the Routing State in comparison with the flat counterpart.

7.5 Conclusions

The analytical analysis estimates the Routing Performance in the H-P2PSIP architecture. In order to have an idea of the efficiency of this architecture, it has been compared respectively a flat CAN overlay network and a Kademlia overlay network with the hierarchical CAN overlay network and the hierarchical Kademlia overlay network based on H-P2PSIP. These comparison is realised since the flat counterparts offer a good starting point for comparisons and we can see the advantages of our proposal with respected to the traditional flat Peer-to-Peer overlay networks. The results obtained give that the obtained Routing Performance is similar in comparison with the canonical format but without the advantages of the hierarchical architecture such as reduction in the Routing State of peer or allow the exchange of information among different overlay networks. The super-peers have to maintain approximately the same Routing State that should maintain a peer in the equivalent flat overlay network. However, the price that it is paid by H-P2PSIP is the fact that all the queries with a destination belonging to other domain must be routed through the super-peers. This analysis is presented in several publications: [MYCGM08], [MYGCM09], [MYBG+08b], [MYBG+08a], [MYBC+09].

Part III

Validation of the designed proposal

Los hombres aprenden mientras enseñan.

Séneca (2 AC-65)

Chapter 8

Validation of H-P2PSIP based on simulation

This chapter validates our H-P2PSIP proposal using a simulation tool. The objective of these simulations is to study the behaviour of our proposal with a large number of peers, which would be the usual usage scenario for our architecture.

8.1 Introduction

In this section, we present several experimental results based on simulation in order to study the performance of a H-P2PSIP architecture. In fact, a hierarchical Kademlia overlay network is used to verify that the Routing Performance and the Routing State meet our expectations. All this work can be found in [MYBG+08b], [MYBC+09], [MYBG+08a]. However, in this chapter it is only presented the work from the last reference, since includes the most detailed simulations with different scenarios and considering churn in peers.

The simulation framework used to validate this works has been the PeerfactSim.KOM[1] P2P network simulator [DMLS04]. This simulator is a packet-level discrete event-based simulator written in Java, which makes the development easier and it can be used in most of the available platforms nowadays. In order to facilitate the simulation of large scale Peer-to-Peer networks, the simulator uses a simple packet latency model between nodes that is the equivalent of the cumulative propagation, forwarding and queuing delay. However, it does not consider some details such as the processing time and the bandwidth of links (links are over-provisioned). In addition, this framework provides an implementation of the Kademlia protocol, thus if a hierarchical Kademlia overlay network is desired, this protocol must be extended to support Hierarchical-ID's and make that some peers take the role of super-peers maintaining two routing tables as it has been discussed in chapter 6.

[1] http://peerfact.kom.e-technik.tu-darmstadt.de/

Finally, considering that a simulation framework is being used, it necessary to validate correctly the obtained results. Therefore, it is necessary to contrast statistically these results. 95% confidence intervals are used to contrast the results and the number of simulations is setup to have an error smaller than the 10% in these confidence intervals.

8.2 Simulation Setup

To run the experiments, we implemented a prototype of the hierarchical Kademlia protocol and a network scenario generator on top of the simulator engine. The objective was to generate Peer-to-Peer network models similar to the behaviour of real life Kademlia peers. For this we assumed network scenarios with an average number of peers between 400 and 10000 and the following number of domains: 1 (i.e. a pure Kademlia network), 5, 10 and 20. The peers were uniformly distributed among the domains. In addition, each domain has a super-peer that facilitates the connection of the domains through the Interconnection Overlay. Only one super-peer ($S_k = 1$) is placed since the Routing Performance penalty is marginal as has been explained in Sec.7.2 and the complexity of the simulation increases a lot. Additionally, the stability of super-peers can be assured as in Skype [RMM08] with some mechanism like [BYGM03], [MHC06] and [MCKS03]. The management of the super-peers is not included in the study and it constitutes future work. Thus, we do not consider churn in super-peers and only churn in peers is applied. This scenario with stable super-peers fits in situations where SP's or ISP's consider the use of Peer-to-Peer applications to distribute the information of their services through their own customers. The motivation is to reduce the transit costs of their traffic that they need to pay to their ISP's. Therefore, they are interested in maintaining themselves super-peers with a high availability since they are necessary for the correct behaviour of the overlay network. The benefit comes because the maintenance of this equipment is smaller than the money saved by reducing their traffic.

Each peer executes four types of operations: joining when it attaches itself to the Peer-to-Peer overlay network, storing a key-value pair, looking-up when searching for a previously stored key in the attempt to find the value and leaving. In order to have scenarios closer to reality, we used an existing study of the KAD implementation of Kademlia [SENB07c] that measures the peer behaviour in terms of churn rate and up-time distributions. Their findings conclude that in a file-sharing KAD network peers arrive and leave with a negative binomial distribution, while the peer session time is similar to a Weibull distribution. Additional details can be found in [SENB07b], [SBEN07] and [SENB07a]. This setup can be considered as a medium-high churn rate scenario since the KAD network is used in eMule and Bittorrent applications where the churn is not at all negligible. Thus, our scenario is a worse case study in comparison with multimedia applications like Skype [BS06], [GDJ06a], [RMM08].

Due to the simulation constraints (such as simulation duration, required computing resources, etc.), each simulation scenario has two phases. The first is a transitory phase where the total number of peers reaches the average targeted in each scenario. This phase does not consider the KAD peers behaviour, since in a real KAD network the arriving and the leaving rates are the same. In the second phase, the peers join and leave the Peer-to-Peer network at the rate given in [SENB07c] with a negative binomial distribution (approximately one peer

every two seconds). In this phase, the average number of peers in the network is the number of peers at the end of the first phase. Because the results from the KAD study were given for a flat Kademlia network, in the hierarchical case, arriving peers are randomly assigned to any of the existing domains with a uniform distribution.

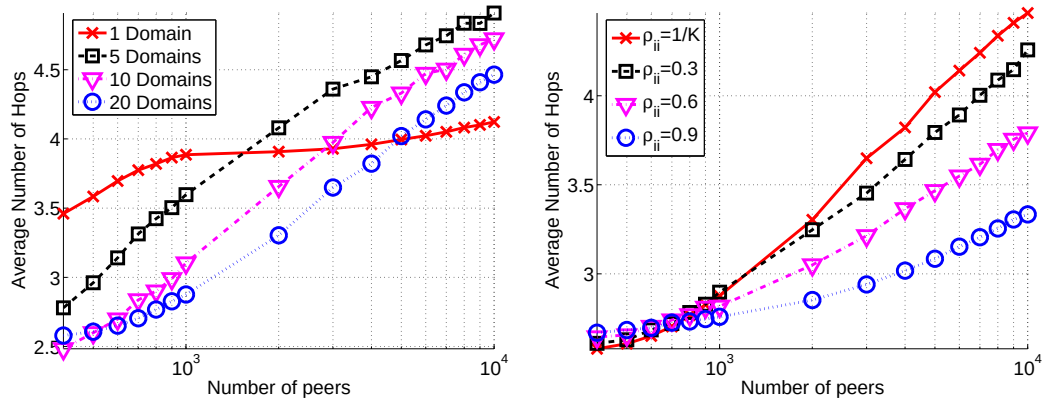
During a session, each peer performs a store operation that is the equivalent to storing its own URI in the Peer-to-Peer network, and a number of look-up operations that are the equivalent to searching for the URI of other peers. Assuming that the look-ups follow the behaviour of the user contacting other peers, we used a Poisson distribution to model them, at an average rate of one call every ten minutes. The transitory first phase was limited to 30 minutes, while the stationary second state spanned up to two hours. As in Kademlia, a maintenance operation was run by each peer every hour after their arrival, in order to refresh their routing tables and republish stored values to neighbour peers. Measurements were taken only during the second phase.

In relation with the setup of the Kademlia overlay, the protocol has been configured with $B = 2^b = 2$, $k = 20$ and $\alpha = 1$. The reason for using $\alpha = 1$ is to facilitate the comparison with other overlays that cannot easily parallelize their operations. Determining the performance for higher values of α is planned for future work. The value of k is used for the size of the buckets and also for the number of replicas of each item inside the overlay.

8.3 Routing Performance

The Routing Performance is calculated for both *node look-up* (search of nodes, replication cannot be used) and *value look-up* (search of resources, replication can be used) operations. The former are the result of the maintenance operations (refresh of the routing tables) and they are performed solely inside the domain or inside the Interconnection Overlay between super peers. The latter are modelled based on peer behaviour of searching for stored values and can span two different domains. In addition, since the value look-ups take advantage of key-value replication, we expect the value look-ups to have a better performance than expected. These operations finish as soon as a key is found. According to the analytical model and considering the assumptions on the simulation, the Routing Performance is estimated using the equations on chapter 7.

Figure 8.1 illustrates the Routing Performance for value look-up operations. In Figure 8.1(a), we have the obtained Routing Performance for 1, 5, 10 and 20 domains. The dependency is logarithmic with the number of peers in a domain (linear on a logarithmic scale), when the number of super peers is kept constant. Because the increase is almost constant while the number of super peers doubles, the result proves the logarithmic dependency of the Routing Performance with the size of the Interconnection Overlay. The number of hops is bounded by Eq.8, which is a constant since it only depends on N. The obtained results are smaller than the theoretical limit due to the replication of the information to combat the churn rate. Additionally, in Figure 8.1(b), we have the Routing Performance for 20 domains and ρ_{ii} equals to $\frac{1}{K}$, 0.3, 0.6 and 0.9. It can be appreciated how the Routing Performance increases as ρ_{ii} increases since the increment of ρ_{ii} makes larger the number of intra-domain



(a) Random Independent Queries with several domains

(b) 20 domains and $\rho_{ii} > \rho_{ij}$

Figure 8.1: *Routing Performance for value look-up operations*

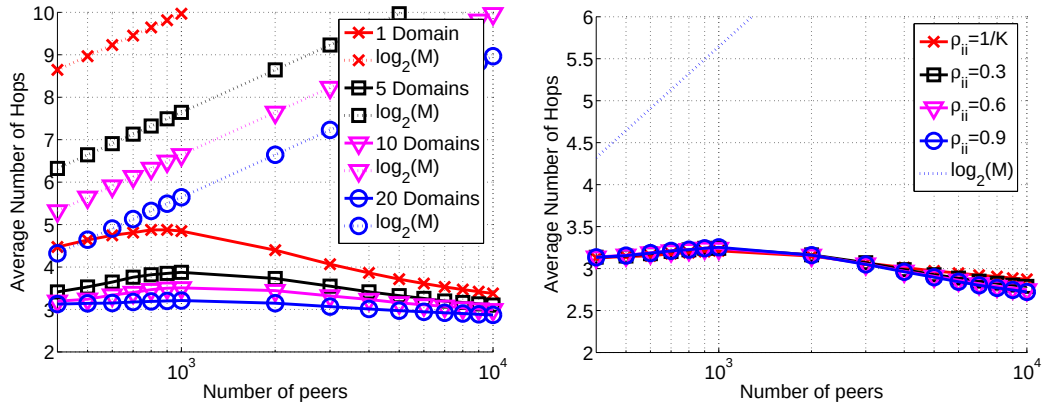
look-ups. This difference is especially relevant when the number of peers is large. We have simulated up to ten thousand peers among all the domains and in a real scenario is expected that this number will be much bigger.

Figure 8.2 shows the Routing Performance for node look-ups for intra-domain operations. They are important because node look-ups for specific nodes are the worst case in comparison with value look-ups since they take advantage of replication. This difference is higher when the number of peers inside the domain is comparable to the replication parameter ($k = 20$) and becomes negligible when the number of domain peers is large enough for the replication to have an important effect. In Figure 8.2(a), we can see how the Routing Performance is smaller than the theoretical ($\log_2 M$ that is for the worst case). As expected, we find in 8.2(b) that ρ_{ii} does not affect to Routing Performance of the node look-ups because is a parameter that defines how many queries are inter-domain and it only affects to value look-ups.

In order to see how good is the analytical analysis on estimating the upper bound of the Routing Performance we have obtained the worst case for the simulate value look-ups in Figure 8.3. We can appreciate in Figure 8.3(a) how the worst cases are close to the upper bound given in chapter 7 and how this bound has a logarithmic dependency. Furthermore, in Figure 8.3(b) we can see how this worse case is independent of ρ_{ii}.

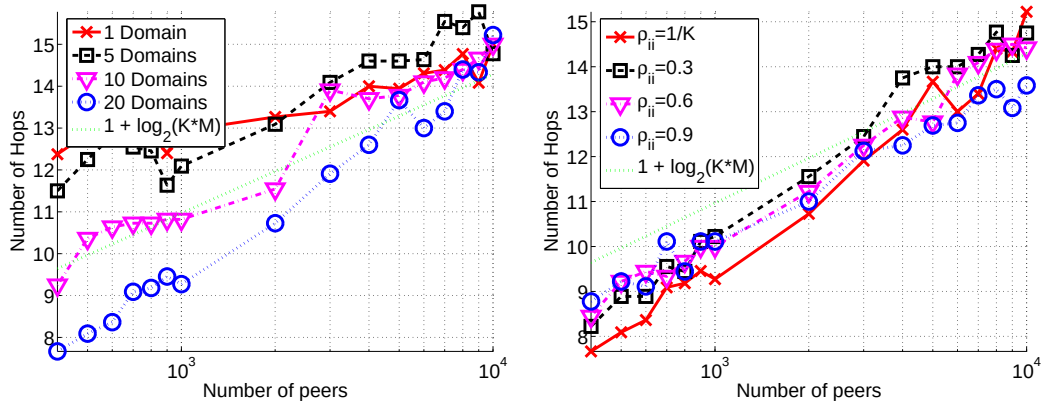
8.4 Routing State

The evaluation of the Routing State intends to determine whether the average number of routing entries maintained by the peers lies within the expected ranges and to illustrate the behaviour of the Routing State when the number of domains changes. For this, we examine the routing tables used for routing inside domains.



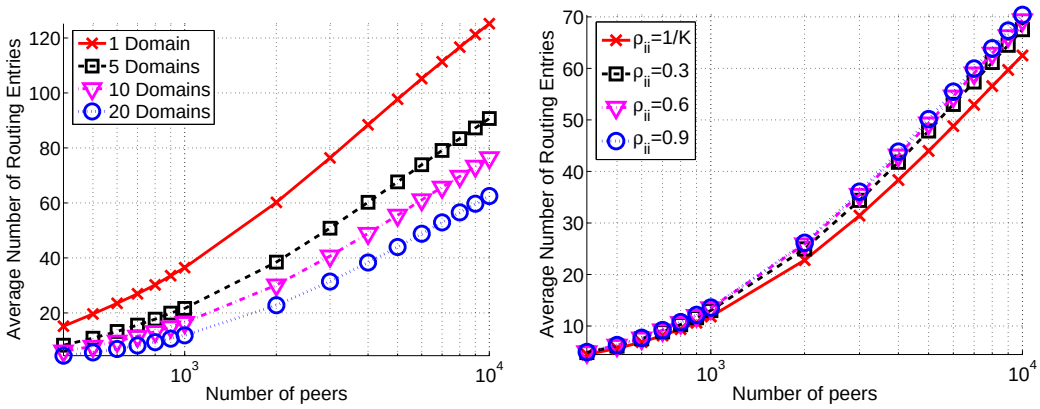
(a) Random Independent Queries with several domains (b) 20 domains and $\rho_{ii} > \rho_{ij}$

Figure 8.2: *Routing Performance for node look-up inter-domain operations*



(a) Random Independent Queries with several domains (b) 20 domains and $\rho_{ii} > \rho_{ij}$

Figure 8.3: *The worst case of Routing Performance for value look-ups operations*



(a) Random Independent Queries with several domains

(b) 20 domains and $\rho_{ii} > \rho_{ij}$

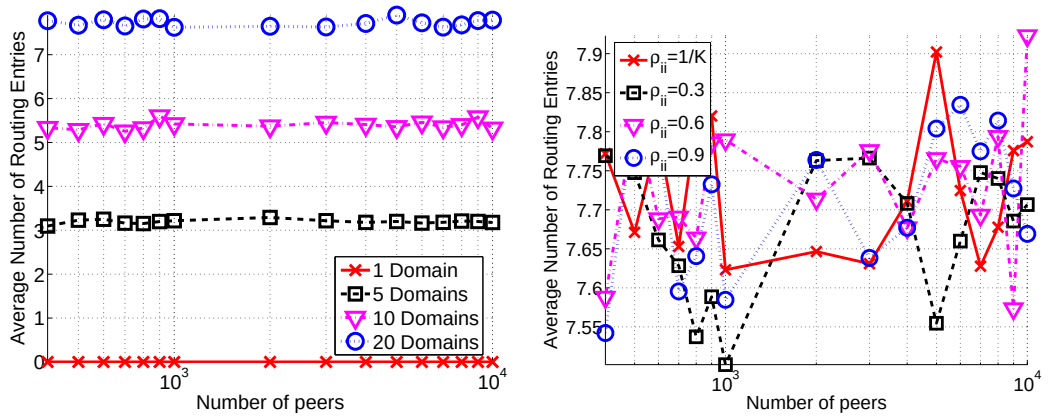
Figure 8.4: *Routing state for intra-domain routing tables*

Figure 8.4 shows the obtained results. We have that: $NE \in [\log_2 N, k \log_2 N]$, where NE is the average number of routing entries. In addition, we can observe a slight dependency between the number of domains and the value of the Routing State in Figure 8.4(a). Since the Routing State is determined solely by the interaction between peers, the explanation for this dependency is the fact that the simulation scenarios use the same number of value look-up operations. In general, the value look-ups are originated in one domain and usually terminated in another domain. However, if the number of domains is small the number of operations that originate and terminate in the same domain increases and consequently the number of routing entries also increases according to the standard Kademlia protocol to populate the bucket entries. Node look-ups cannot influence the Routing State because they take place only inside a domain and have no relationship to the number of domains. We can also see how there is also a little dependency with ρ_{ii} in Figure 8.4(b). If ρ_{ii} is large, the intra-domain queries are more likely and the intra-domain routing tables are slightly more populated.

As expected, the number of hops needed in the Interconnection Overlay (see Figure 8.5(a)) is roughly the same for any number of peers, since it only depends on the number of domains, K. In addition, the logarithmic dependency with K can be observed through the large increase in the number of hops from one domain to five domains and the same increase between 5, 10 and 20 domains. Furthermore, we can see how this value is independent with ρ_{ii} in Figure 8.5(b).

8.5 Conclusions

A great number of simulations were done using PeerFactSim.KOM. These simulations consist in a hierarchical Kademlia overlay network. The simulations are as realistic as possible. In order to achieve this objective, the join and departure rates as well as the churn



(a) Random Independent Queries with several domains

(b) 20 domains and $\rho_{ii} > \rho_{ij}$

Figure 8.5: *Routing state for Interconnection Overlay routing tables*

patters are based on measurements published in research papers (see section 8.2). The obtained results are interesting if we consider the results related with the Routing Performance are better than expected. One reason for this improvement, it is clearly motivated for the use of replicas as redundancy mechanism to fight against the undesirable churn rate. In addition, the Routing State is in the interval expected according to original Kademlia paper, which gives that the routing maintenance algorithms are behaving as expected. Thus, the obtained results cover the expectations of the analytical model. These simulations are presented in [MYBG+08b], [MYBC+09], [MYBG+08a]. The last one presents the most complete simulations, which are used to present this chapter.

Inteligencia militar son dos términos contradictorios.

Groucho Marx (1890-1977)

Chapter 9

An implementation of H-P2PSIP and its validation based on Modelnet

This chapter gives an overview of the obtained results based on an implementation of H-P2PSIP. This implementation is based on the P2PP protocol, since no RELOAD implementation is available yet. P2PP is the most important precursor of RELOAD and it has many of its functionalities. Therefore, the results in this chapter have are relevant since they offer an implementation of H-P2PSIP using a real TCP/IP stack.

9.1 Introduction

After providing a mathematical analysis of the H-P2PSIP architecture and checking its behaviour in the PeerFactSim.Kom simulator, the next step in our work is the development of a real implementation of our proposal. The objective of this implementation is to go one step further in comparison with the simulations and to evaluate its behaviour in real conditions. The added value of this development is the study of our proposal over a real TCP/IP stack with conditions as close as possible to a real deployment in the current Internet.

9.2 Implementation

The developed implementation is an extension of a previous existing deployment. After a deep search of different candidates, the starting point of our work is the Peer-to-Peer Protocol[1] (P2PP) [BSM07], [Coh08] from the University of Columbia. Several candidates[2] were available when this implementation was chosen, but P2PP was selected since it was one of the reference protocols for the current definition of RELOAD [JLR+09a] in the P2PSIP WG

[1] http://www1.cs.columbia.edu/ salman/peer/

[2] http://www.p2psip.org/implementations.php

and it was also one of the most mature and better designed implementations; thus, its use is completely justified.

We are not going to go into the details of the developed implementation; however, it is interesting to comment what actions has been realised to support a hierarchical architecture such as H-P2PSIP if a protocol for flat overlays networks is used as starting point. The main points that we needed are the following:

- *Hierarchical-ID support:* the original design only supports legacy flat overlay networks. Therefore, it is necessary to add the feature of Hierarchical-ID's to support H-P2PSIP.

- *Prefix-ID management in peers:* it is not only necessary to include the Hierarchical-ID support, it is also necessary to add the logic that manages this Hierarchical-ID. With this change, peers have to check if a query belongs to its own domain; if that is not the case, the query must be forwarded to a super-peer.

- *Super-peer support:* the super-peer role does not exist on a flat structured overlay network; thus, it is necessary to define it since is crucial for our design. Two functions are especially important in a super-peer:

  - *Support of both levels of the hierarchy:* a super-peer must be attached to its own overlay and also to the Interconnection Overlay
  - *Forwarding of queries:* if a super-peer receives a query that belongs to other overlay, it has to forward the petition to the super-peer that belongs to the destination overlay network.

9.3 Scenario Setup

A scenario has been setup for the evaluation of our implementation of H-P2PSIP over a real TCP/IP stack. The main requirement was that the scenario must be useful for evaluating the final release of our implementation but it must be also capable of being flexible enough for the deployment and debugging of the code. Several options were considered such as Modelnet[3] [VYW+02], Emulab[4] [HRS+08] and Planetlab[5] [CCR+03]. Emulab and Planetlab use an infrastructure that requires the participation of other institutions; therefore the deployment would depend on external partners; this situation is not desirable for the moment, especially if new code is being created and debugged. On the other hand, Modelnet allows a complete control of the infrastructure to create the desired scenario. In addition, it also allows repeatability of experiments under the same conditions, which it is especially desirable for debugging. Therefore, our scenario has been based on Modelnet.

[3] https://modelnet.sysnet.ucsd.edu/
[4] http://www.emulab.net/
[5] http://www.planet-lab.org/

9.3.1 Modelnet Setup

The Modelnet scenario is built considering the roles that a node has in this infrastructure. Two roles are possible: *edge nodes* and *core nodes*. The core nodes take care of emulating any networking topology; an XML file is used to define the links, the connections among them and their characteristics such as bandwidth, delay or probability of error. On the contrary, the edge nodes are machines that are connected to the core network emulated by the core nodes. In our experiments, we run 1000 edge nodes and each one runs one instance of our implementation. In order to have a platform as much scalable as possible, the edge nodes are executed in a virtualization platform based on OpenVZ[6]. This entire infrastructure is executed with five machines with 16 GB of RAM and QuadCore processors.

In order to create a scenario as real as possible, the dataset provided by the ARK[7] project in CAIDA[8] is used. The XML that defines the emulated core network is based on the information provided by this project; thus, the emulated network topology is a reasonable representation of the current Internet.

9.3.2 Peers setup

It is necessary to include in the scenario the peer behaviour in Peer-to-Peer networks. This is a difficult task and it is necessary to take into account previous work. The session time distributions, and churn rate of peers have been configured considering the measurements in [SENB07c], [SBEN07] and [SENB07a]. Nevertheless, some modifications with respect these measurements have been done. The obtained results in the previous cited references is based on a population of peers much larger than our 1000 nodes; thus, if we use the measured churn rate values, we find the churn rate is really high in comparison with the population of our experiment. Therefore, we settled in a churn rate that is proportional to our number of peers taking as reference the previously cited papers. In relation with the queries, a Poisson distribution is used.

On the other hand, the Kademlia parameters of P2PP are the same ones of the Kademlia overlay network used in the simulations. These parameters are $B = 2^b = 2$, $k = 20$ and $\alpha = 1$.

9.3.3 Experiments setup

The experiment is divided in two phases. First, we have a transitory phase limited to 30 minutes, during this time the average number of peers needed to realise the experiment join the hierarchical overlay network. The second phase, it is the stationary state and it has a duration of 50 minutes. In this phase, the negative binomial distribution for joins and departures of peers as well as the Poisson distribution for the queries are applied. During

[6] http://wiki.openvz.org/Main_Page/
[7] http://www.caida.org/projects/ark/
[8] http://www.caida.org/home/

this phase, the data are collected and presented in this chapter.

Therefore, once that a description of the configure scenario has been described, we can present the obtained results. These results have been obtained by repeating the experiments several times and calculating the associated 95% confidence intervals for these measurements.

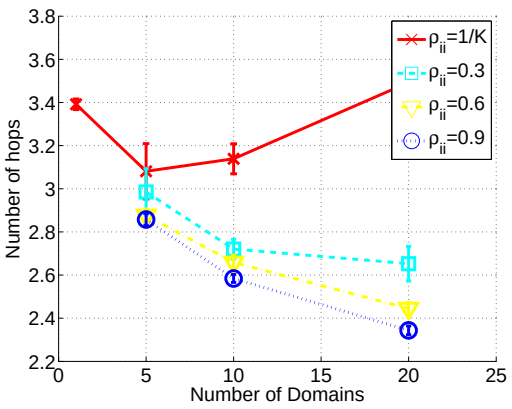
The importance of these results must be highlighted since these values have obtained using a TCP/IP stack and the core network emulation is based on real measurements from the ARK project of CAIDA. Therefore, these results are expected to be close to a real implementation running on the Internet.

9.4 Results with random localization of peers

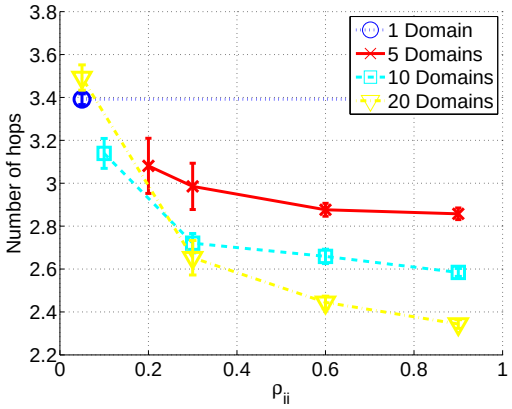
This first set of results is based on the random placing of peers in some of the locations provided by the ARK project of CAIDA. The objective of this set of experiments is to see how a hierarchical Peer-to-Peer overlay network will behave when the peers are placed randomly in the different locations (countries) as provided by the ARK project.

9.4.1 Routing Performance

The average number of hops needed to reach the desired destination is plotted in Figure 9.1. Figure 9.1(a) shows the number of hops made to reach the destination with respect to the number of domains that have been use to group the peers; this figure also shows these number of hops for different values of the intra-domain hit probability (ρ_{ii}). ρ_{ii} is the probability of a peer to perform a query inside its own domain instead of any other external domain. The first important point is the fact that the expected number of hops is close and slightly better with respect to the simulations in [MYBC+09] and [MYBG+08a]. When the queries are randomly made among the different domains ($\rho_{ii} = 1/K$, the red continuous line), we observe how the number of hops starts to be smaller in comparison with the flat counterpart (first point of the plot that corresponds with $K = 1$). As the number of domains increases, the number of hops increases slightly but very close with respect to the flat counterpart Routing Performance. This increment is produced since $\rho_{ii} = 1/K$ and the probability of looking in other domain increases which implies an extra number of hops (one hop to reach the super-peer plus the number of hops in the Interconnection Overlay). On the other hand, we can observe, while the intra-domain hit probability increases ($\rho_{ii} > 1/K$), how the number of hops decreases since the extra hops to reach other domains are not necessary. In order to have a better view, Figure 9.1(b) shows the number of hops made to reach the destination with respect to the intra-domain hit probability. This plot gives a better view of the effects of the intra-domain hit probability. If the intra-hit domain probability increases, more queries are performed on the own domains of each peer and the average number of hops is smaller. Furthermore, the plot gives a better comparison of the Routing Performance with respect to the flat counterpart since it is clearly differentiated with a dotted blue line. We can see how the number of hops is never bigger than the equivalent flat overlay network.

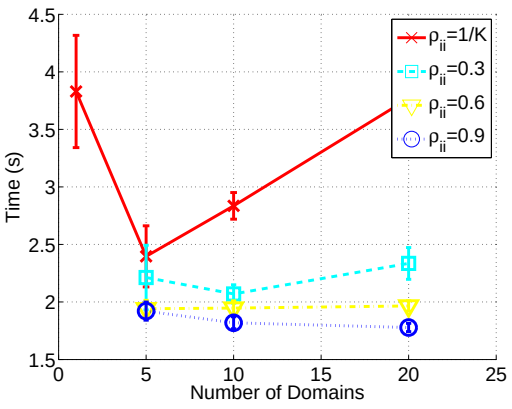


(a) With respect to the number of domains

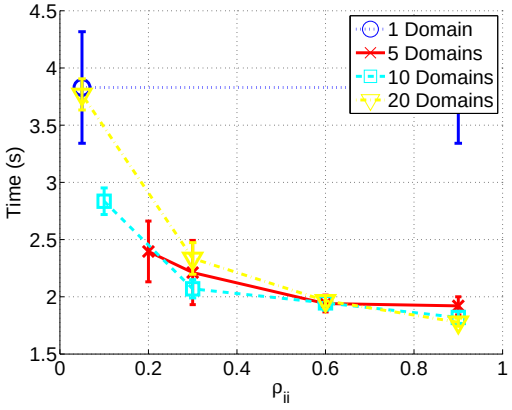


(b) With respect to the probability of querying the own domain

Figure 9.1: *Average number of hops with random geolocated peers per domain*



(a) With respect to the number of domains



(b) With respect to the probability of querying the own domain

Figure 9.2: *Average Delay Time with random geolocated peers per domain*

In relation with Figure 9.1, we have also Figure 9.2 that shows the average delay suffered by the queries in our scenario. Figure 9.2(a) represents the delay with respect to the number of domains and the figure includes the legend for different values of ρ_{ii}. We can see how a certain correlation exists among the shapes of this Figure 9.2(a) and Figure 9.1(a). When $\rho_{ii} = 1/K$, the delay increases if the number of domains increases. This effect is caused by the extra number of hops needed to reach the information in other domains (Figure 9.1(a)). On the other hand, if ρ_{ii} increases, the average delay is considerably reduced since the number of queries to other domains is lowered and fewer hops are needed. Additionally, Figure 9.2(b) shows how the delay depends more on ρ_{ii} rather than on the number of domains; only appreciable differences exist if ρ_{ii} has small values. Again, the delay is never larger that the equivalent flat overlay network counterpart.

In general, we see how the hierarchical architecture with different parameters is more efficient, from the point of view of Routing Performance, rather than the flat counterpart.

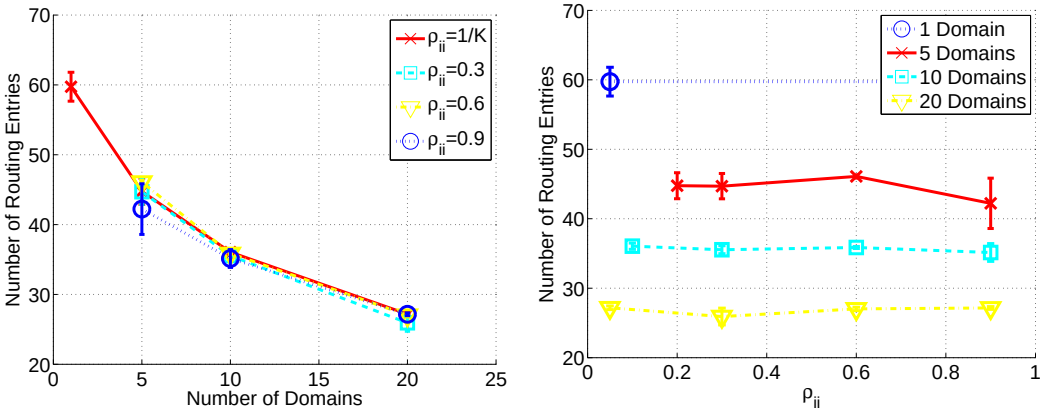
9.4.2 Routing State

In the previous section, the Routing Performance and delay were analysed; nevertheless, it is also interesting to study which is the load that peers must sustain to support the structure of the overlay network in order to correctly route the queries. We have collected the average number of routing entries of the peers in order to estimate the average information that must be maintained by each one. It must be highlighted that these routing entries not only consume memory on the peers, but also must be updated to assure that they are up-to date. Therefore, the number of entries in the peers also partially reflects the effort that must be maintained by the peers to assure the validity of their information.

Figure 9.3 presents the average number of entries in the peers. Concretely, Figure 9.3(a) shows how the number of needed routing entries decreases if the number of domains increases. This effect is expected due to the limitations of our testbed; thus, we must maintain a constant number of peers in our experiments. Thus, if the number of domains increases, the number of peers per domain decreases and consequently the number of routing entries per peer. On the other hand, Figure 9.3(b) demonstrates how the intra-domain hit probability (ρ_{ii}) has a negligible effect on the number of entries that are needed. The overlay routing tables must be maintained with independence of this parameter and they depend on the number of peers in an overlay network.

The number of routing entries in Figure 9.3 is larger than the values obtained in the simulations. The explanation of this mismatch is in the slightly differences among the implementation of the Kademlia protocol in simulator and P2PP. However, the obtained results can be considered valid since the number of routing entries is among the theoretically expected values [MM02].

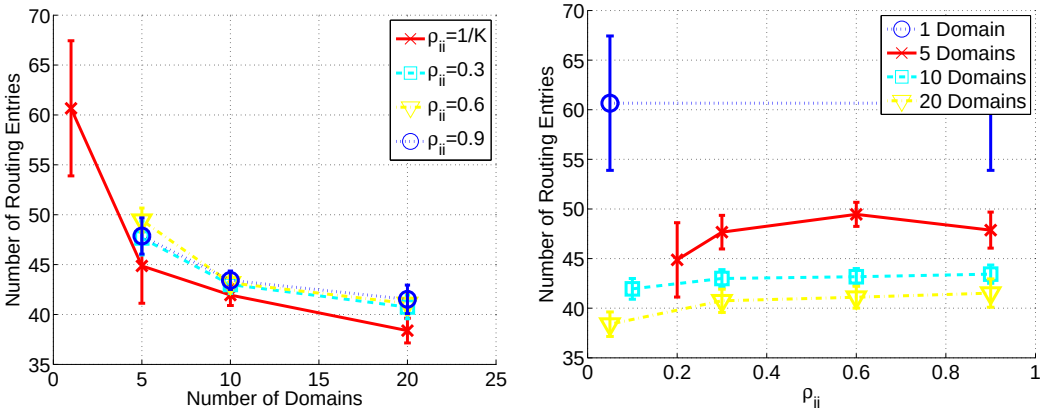
Furthermore, in addition to the previously commented results, we have also Figure 9.4 where the Routing State information of super-peers is shown. The super-peers, in addition to maintaining their own domain overlay, must also maintain the overlay routing table associated to the Interconnection Overlay. Figure 9.4 shows the average on routing entries



(a) With respect to the number of domains

(b) With respect to the probability of querying the own domain

Figure 9.3: *Routing State in peers with random geolocated peers per domain*



(a) With respect to the number of domains

(b) With respect to the probability of querying the own domain

Figure 9.4: *Routing State in super-peers with random geolocated peers per domain*

taking into account both routing tables. The size of the overlay routing table that belongs to the own domain is similar to the given in Figure 9.3. Therefore, the extra number of entries corresponds to the overlay routing table of the Interconnection Overlay. The difference is not very large since the number of interconnected domains is small due to the limitations of our testbed; we cannot run more than 20 different domains and more than 1000 peers. Again, we can see in Figure 9.4(a) how the number or routing entries decreases if the number of domains increases and it is also found how the intra-domain hit probability negligibly affects these values in Figure 9.4(b).

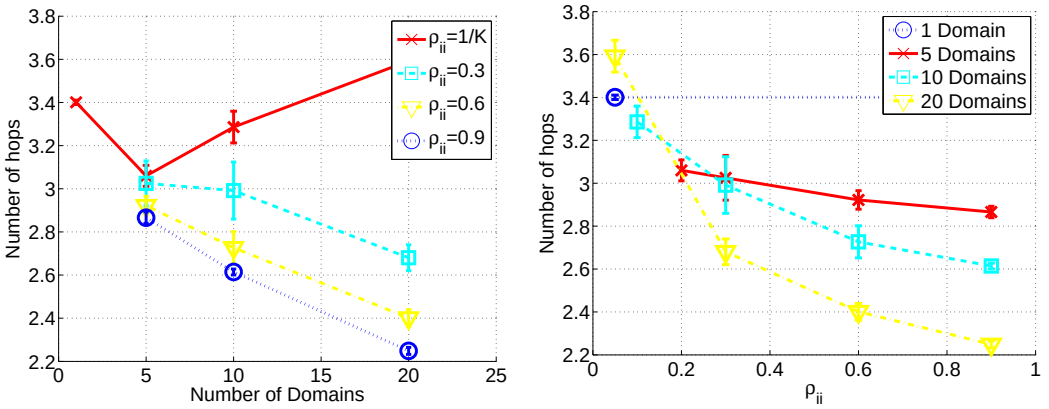
9.5 Results with aggregated peers according to their geolocalization

This second set of results takes into account how the local aggregation of peers can contribute to improve the efficiency of the system. This proposed aggregation is based in the geographical locations of peers, each domain built in these experiments has all the peers located in the same country. The emulated core network is the same one used in the previous section. According to the CAIDA dataset the delay among peers of the same country is smaller than the delay among peers of different countries.

9.5.1 Routing Performance

Figure 9.5 presents the average number of hops that are needed to find the desired resources in the overlay network. We can see how the results are very similar with respect to Figure 9.1 that contains the results for randomly localized peers. This makes sense since the number of hops depends only on the number of peers in the overlay network. An appreciable difference exists in the point with 10 domains and $\rho_{ii} = 0.3$. The explanation to this behaviour is the fact that in some of the experiments some bad conditions have occurred and the Routing Performance has been worse than expected. In fact, we can observe in Figure 9.5(b) how the confidence interval is the largest in comparison with the other ones. However, the results can be considered valid since the error is still under the 10%.

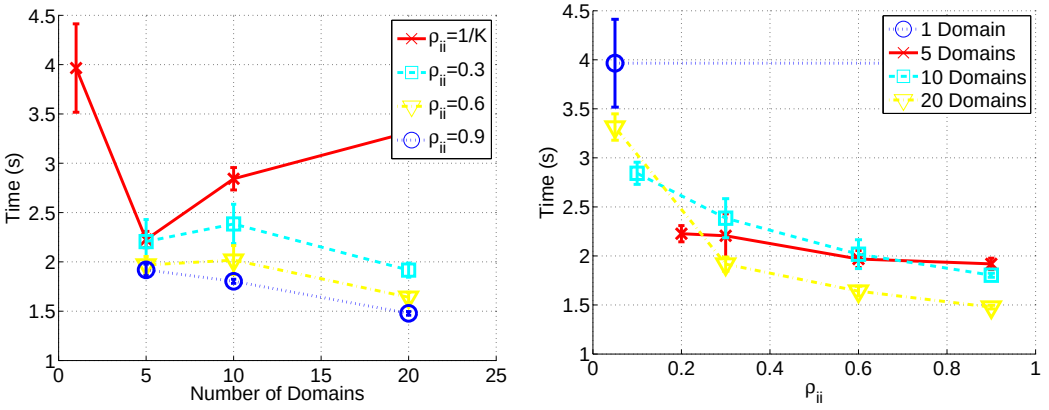
In addition to the number of hops, we also have the average delay spent in the queries. This information is provided in Figure 9.6. If it is compared with Figure 9.2, the similarities are quite evident. The expected behaviour would be a smaller delay since the peers are aggregated in the same localizations. However, this effect is negligible and it is only appreciated when the number of domains is 20. This can be seen in Figure 9.6(b), where the yellow dashed line (20 domains) is below the other ones. The explanation to this small improvement is our number of peers in the experiment. Since we have the peers divided in domains, the number of peers per domain is not so large, in fact the number of hops to reach the destination is also small (between 2 or 3 hops). Thus, the margin for improvement is small. If the number of hops would be bigger, the reduction in the delay would be more evident.



(a) With respect to the number of domains

(b) With respect to the probability of querying the own domain

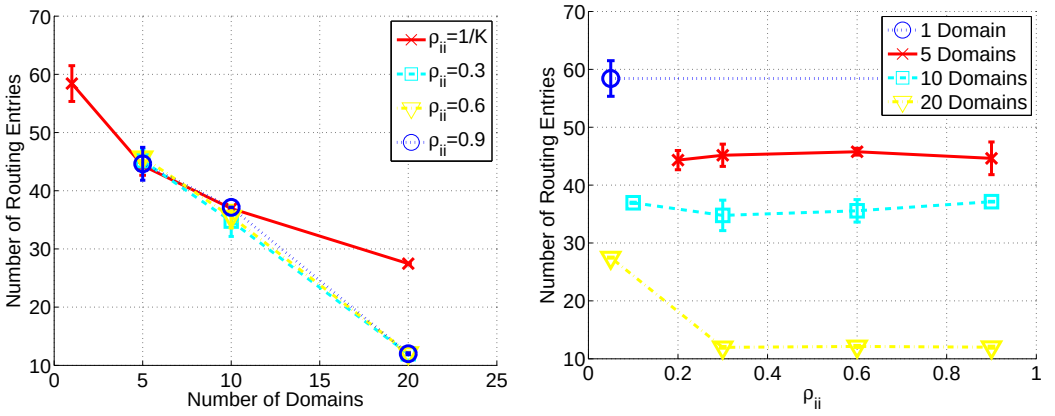
Figure 9.5: *Average number of hops with peers per domain geolocated in the same country*



(a) With respect to the number of domains

(b) With respect to the probability of querying the own domain

Figure 9.6: *Average Delay Time with peers per domain geolocated in the same country*



(a) With respect to the number of domains

(b) With respect to the probability of querying the own domain

Figure 9.7: *Routing State in peers with peers per domain geolocated in the same country*

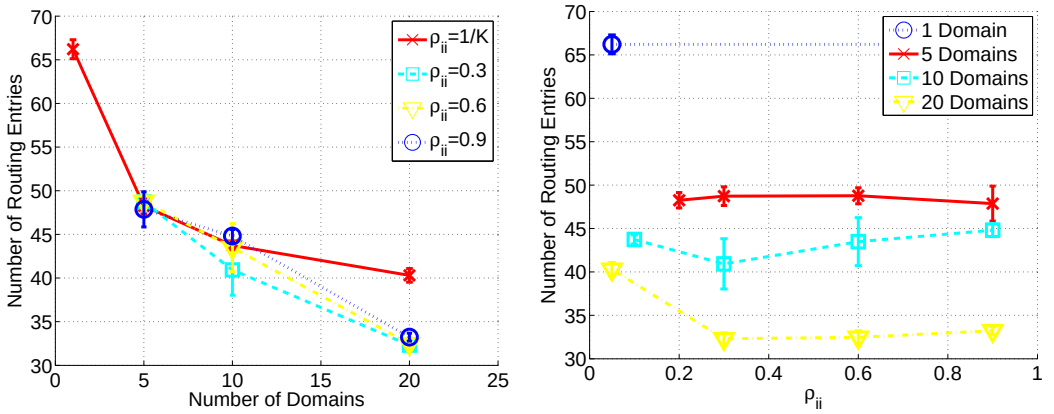
9.5.2 Routing State

Figure 9.7 presents the size of the routing tables in the peers of each domain. We can see again that the results are very similar with respect to the case of randomly localized peers (Figure 9.3). Nevertheless, a remarkable difference can be appreciated in Figure 9.7(b). This figure shows how the size of the routing tables is significantly smaller when the number of domains is 20 and $\rho_{ii} > 1/K$. This behaviour is difficult to understand and its study is complex. We think it is caused by the way that Kademlia updates its routing tables. This effect can be produced by cross effects among the mechanism to update the buckets, the frequency which is used to update the buckets (depends on ρ_{ii} since the routing tables are adapted with the information retrieved when a query is done) and the better network conditions that it has the geolocated scenario. We are studying this behaviour deeply to discover why this effect happens.

Finally, Figure 9.8 shows the size of the routing tables in the super-peers. We can observe the same effects as in the peers. However, we observe how the number of routing entries is larger because of the routing entries required for the Interconnection Overlay. The decrement of the overlay routing table for 20 domains and $\rho_{ii} > 1/K$ observed in Figure 9.7 can be also appreciated in this figure.

9.6 Conclusions

This chapter presents the implementation of our H-P2PSIP proposal. This implementation is based on the P2PP protocol since there is no available implementation of RELOAD. P2PP is the precursor of RELOAD and it is quite similar in the key points that make RELOAD interesting. Therefore, we can consider the obtained results valid. The evalua-



(a) With respect to the number of domains

(b) With respect to the probability of querying the own domain

Figure 9.8: *Routing State in super-peers with peers per domain geolocated in the same country*

tion of this implementation is based on an experiment with 1000 nodes running our implementation and the core network is emulated with the Modelnet software. Modelnet offers an interesting infrastructure that allows repeating experiment under very similar conditions. In order to have representative results, the emulated core network is built from the date set provided by the ARK project from CAIDA. This data set provides the characteristics of the different links that exist among different countries and it is a reasonable abstraction of the current Internet. Two scenarios have been used, the first one consists on random geolocalization of peers inside each domain. On the other hand, in the second one, the peers inside a domain are geolocated in the same region (country). The obtained results are very close to our previous simulations [MYBC+09] and [MYBG+08a] and they are slightly better in terms of Routing Performance and worse in terms of Routing State. This is reasonable since more routing entries imply fewer hops. If the number of domains or the intra-domain hit probability increases, the Routing Performance and delay decreases if $\rho_{ii} > 1/K$. In addition, the obtained results are smaller than the flat counterpart is. The size of the routing tables although is higher than the obtained in the simulations is among the theoretically expected values [MM02]. These differences are caused because the Kademlia implementation in the simulator is not the same as in the P2PP protocol.

Therefore, the functionality of the proposal is demonstrated with a results close to our expectations and we can conclude the validation of H-P2PSIP.

Part IV

Conclusions and future work

Nunca he encontrado una persona tan ignorante de la que no pueda aprender algo.

Galileo Galilei (1564-1642)

Chapter 10

Conclusions

This Thesis has presented the design of a hierarchical architecture, named H-P2PSIP that allows the exchange of information among different structured overlay networks. Any DHT, independently of its topology, characteristics or configuration parameters can participate in the hierarchical overlay network that enables this exchange of information. A summary of the design is given in the next lines. First of all, we define a hierarchical space of identifiers and each peer or resource in the hierarchical overlay network has a Hierarchical-ID. This Hierarchical-ID is composed by a Prefix-ID and a Suffix-ID. The Prefix-ID represents the overlay where a peer or resource is attached and it is used to route the queries to correct domain inside the hierarchical DHT overlay network. On the other hand, the Suffix-ID represents the peer or the resource itself and it is used to route the queries inside each domain. To support the exchange of information it is necessary to make use of super-peers. These super-peers belong to their own overlay but they also participate in the Interconnection Overlay. Therefore, the super-peers are the support of a hierarchical overlay network with two levels of hierarchy. The Interconnection Overlay acts very similar to a DNS resolver but in the Prefix-ID domain. All super-peers store their associated information in the Interconnection Overlay; thus, if one query wants to be forwarded to another domain, the super-peers can obtain the contact information of other super-peers using the Interconnection Overlay. This simple mechanism allows the exchange of information among different overlay networks and also gives the flexibility of running any DHT in each different domain.

It is necessary now to verify if this design fulfils the desired requirements (seen in the Introduction, Chapter 1) and goals (Chapter 5) that was originally required for this Thesis:

- *Creation of a General Hierarchical DHT Overlay Network:* the adoption of a Hierarchical-ID splits the routing on each domain with respect to the routing in the Interconnection Overlay. They are completely independent and this fact allows that *any* DHT can be used in *any* of the domains in the lower layer of the hierarchy as well as the most suitable overlay network, depending on the usage scenario and requirements, can be used in the Interconnection Overlay. Therefore, we can positively say that the requirement of a general hierarchical DHT overlay network is fulfilled. The

information related with this topic can be found in chapter 6.

- *Exchange of information among different overlay networks:* our architecture is based on super-peers. They take care of forwarding the queries among different overlays in order to get any desired information. Furthermore, a common payload and signalling are used. This fact assures that the exchange of resources among different overlay networks, although they maintain different topologies, is possible.
  - *The overlay networks can have different topologies:* since the routing based on the Prefix-ID is independent with respect to the routing based on the Suffix-ID, any topology can be used to realize this routing (Kademlia, Chord, ...).

  The details are in chapter 6.

- *Usage of standards:* we have adopted the RELOAD protocol from the P2PSIP WG. This protocol is designed to support the implementation of any DHT overlay network. Its adoption assures a standard mechanism that defines a common payload that allows the exchange of information among any peer as well as a common format for the exchanged data structures. However, it is necessary to accommodate the proposal of our Hierarchical-ID to the current proposal of RELOAD. Our understanding considers that the best way to proceed would be to allow a variable length in the Peer-ID managed by RELOAD. This modification will allow accommodating easily any Hierarchical-ID, although other solution could be also applied. A wider discussion can be found in section 6.4.

- *Provision of mechanisms to assure the scalability of the solution:* our architecture, based on super-peers, avoids that legacy peers have to add overlay routing information to their tables from peers attached in other domains. Therefore, this assures a higher scalability. On the other hand, the super-peers must maintain two overlay routing tables. In fact, the total number of routing entries is smaller in comparison with the number of routing entries needed if a flat overlay network is maintained among all the peers from all the domains that want to exchange information. A detailed mathematical analysis is provided in chapter 7. However, the real price that we have to pay for having more lightweight peers is the fact that the queries with destination to other domains must always be forwarded and processed by the super-peers. Our experiments have not considered the churn in super-peers since different mechanisms already exist to choose the most suitable super-peers taking into account the heterogeneity in characteristics of the peers participating in the overlay network. However, it would be interesting to study this fact more carefully. This last item indicates that the provision of mechanisms to assure the scalability of the solution is only partially covered.

- *Validation of the proposed solution:* a mathematical analysis (chapter 7) was performed when the proposed solution was designed in order to estimate its performance. The results are interesting since they indicate that the Routing Performance (the number of hops to reach the destination) is similar to the flat counterpart but without the advantage of interconnecting (aggregating) different DHT overlay networks. Furthermore, the size of the overlay routing table in the peers does not increase as it has been

explained before. However, these expectations should be validated and the following mechanisms were used:

- *Simulation:* a discrete time event simulator specialized in Peer-to-Peer networks is used to validate our solution. This simulator has several overlay networks implemented and has been used for realizing a hierarchical Kademlia overlay network. The most important changes in the design have been the support of Hierarchical-ID's and the inclusion of the new super-peer role. The peer behaviour has been done taking into account real measurements published in the literature. The obtained results (validated statistically) are better than expected since the use of replicas, to prevent the effects of churn, improves the Routing Performance. In addition, the size of the overlay routing tables is in the expected range. Detailed comments about these results are provided in chapter 8.
- *Implementation:* in addition to the simulations, we also provide an implementation of our proposal with a hierarchical Kademlia overlay network, the same one used in the simulations. This implementation has been tested in a Modelnet platform where the emulated core network has been constructed taking into account the data set provided by the ARK project from CAIDA. This assures a reasonable representation of the current Internet. The expected performance is quite similar to the results given by the simulator but some slight differences exist. These differences are mainly caused by the differences in the implementation of some details in the Kademlia protocol. These results are deeply studied in chapter 9.

We can see how most of the objectives have been addressed. In particular, in order to properly complete the item related with provisioning mechanisms to assure the scalability of the solution; it may be interesting to define some specific mechanisms for the management of super-peers in DHT overlay networks.

Finally, a summary of the publications, where the contributions of this Thesis have been published, is given. The architectural design of H-P2PSIP is published in [MYBG+08a], [MYBC+09]. The analytical model that estimates the performance of our solution can be found in [MYCGM08], [MYGCM09], [MYBG+08b], [MYBG+08a], [MYBC+09]. The simulation study that verifies H-P2PSIP is detailed in [MYBG+08b], [MYBC+09], [MYBG+08a]. Finally, the results related to our implementation are pending to publish but we expect to do it a short term. Other publications related with Peer-to-Peer networks but focused in other topics are [MYCGM07], [CGNMY07], [CGMYN07].

El futuro tiene muchos nombres.
Para los débiles es lo inalcanzable.
Para los temerosos, lo desconocido.
Para los valientes es la oportunidad.

Victor Hugo (1802-1885)

Chapter 11

Future work

Although there are several contributions related with hierarchical overlay networks and the communications among different overlay networks have been developed, this research topic is not finished and several open issues exist. Thus, it is interesting to comment the possible next steps than can be followed after this Thesis.

- *Simulation based studies with different overlay networks:* our current simulations are based on a hierarchical Kademlia overlay network. Thus, the next step is to consider different overlay networks in the experiments considering that the proposal works and has been compared with its flat counterpart giving good results.
- *Implementation based studies with different overlay networks:* with the same philosophy of the previous item, the next step is to improve our implementation with different overlay networks such as Chord or Bamboo.
- *To perform an implementation based on RELOAD:* it could be considered that the usage of P2PP is not enough, therefore if some implementation of RELOAD would be available, it would be really interesting to implement H-P2PSIP over a final version of the P2PSIP protocol.
- *Improve the validation of H-P2PSIP:* although our validation is based on a real implementation with a real TCP/IP stack, the core network is an emulation of the Internet (information extracted from CAIDA). However, it would be desirable a more realistic approach. One option is to extend our validation to Planetlab.
- *To study mechanisms for super-peer management in H-P2PSIP:* although the literature related with super-peers is large, it is mainly focused on unstructured overlay networks. Therefore, improvements in the management of super-peers can be obtained if we consider the structure that maintains the DHT overlay networks.
- *To contribute on the P2PSIP WG with the lessons learned in H-P2PSIP:* nowadays the P2PSIP WG is focused on defining the basic support for flat overlay networks and

there is an opportunity to contribute on the support of hierarchical architectures or enabling the exchange of information among different overlays when these items will be open, if any.

References

[ALAS05] M.S. Artigas, P.G. Lopez, J.P. Ahullo, and A.F.G. Skarmeta. Cyclone: a novel design schema for hierarchical dhts. In *Peer-to-Peer Computing, 2005. P2P 2005. Fifth IEEE International Conference on*, pages 49–56, Aug.-2 Sept. 2005. 19, 20, 21

[ASB+06] R.L. Aguiar, S. Sargento, A. Banchs, C.J. Bernardo, M. Calderon, I. Soto, M. Liebsch, T. Melia, and P. Pacyna. Scalable qos-aware mobility for future mobile operators. *Communications Magazine, IEEE*, 44(6):95–102, June 2006. 48

[ATS04] S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 36(4):335–371, 2004. 7, 8

[BB06] R. Brunner and E. Biersack. A performance evaluation of the Kad-protocol. Technical report, Corporate Communications Department. Institut Eurécom, 2006. 8, 14

[BMSW08] D. Bryan, P. Matthews, E. Shim, and D. Willis. Concepts and terminology for peer to peer sip, July 2008. Internet Draft draft-ietf-p2psip-concepts-02.txt. 23

[BS06] S. A. Baset and H. G. Schulzrinne. An analysis of the skype peer-to-peer internet telephony protocol. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–11, April 2006. 7, 14, 23, 72

[BSM07] S. Baset, H. Schulzrinne, and M. Matuszewski. Peer-to-peer protocol (p2pp), November 2007. Internet Draft draft-baset-p2psip-p2pp-01.txt. 39, 79

[BWDD02] P. Backx, T. Wauters, B. Dhoedt, and P. Demeester. A comparison of peer-to-peer architectures. In *Eurescom Summit*, volume 2002, 2002. 8

[BYGM03] B. Beverly Yang and H. Garcia-Molina. Designing a super-peer network. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 49–60, 2003. 20, 21, 46, 72

[CCR+03] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, and Mic Bowman. Planetlab: an overlay testbed for broad-coverage services. *SIGCOMM Comput. Commun. Rev.*, 33(3):3–12, 2003. 80

[CGC00] Andrew T. Campbell and Javier Gomez-Castellanos. Comparison of ip micromobility protocols. *SIGMOBILE Mob. Comput. Commun. Rev.*, 4(4):45–53, 2000. 48

[CGMYN07] Ruben Cuevas, Carmen Guerrero, Isaias Martinez-Yelmo, and Carlos Navarro. Bittella: A novel content distribution overlay based on bittorrent and social groups. In *1st International Workshop on Peer to Peer Networks (PPN'07)*, Nov. 2007. 3, 95

[CGNMY07] Ruben Cuevas, Carmen Guerrero, Carlos Navarro, and Isaias Martinez-Yelmo. Bittella: A new protocol for unstructured p2p networks based on the small world per content structure. In *Med Hoc Net 2007*, Corfu, Greece, June 2007. 3, 95

[Coh08] Kai Michael Cohrs. Implementation and evaluation of the peer to peer-protocol *p2pp* for p2psip. Master's thesis, Georg August Universitat Gottingen Zentrum fur Informatik, 2008. 40, 79

[DMLS04] V. Darlagiannis, A. Mauthe, N. Liebau, and R. Steinmetz. An Adaptable, Role-based Simulator for P2P Networks. *Proceedings of the International Conference on Modeling, Simulation and Visualization Methods*, pages 52–59, 2004. 71

[DR06] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.1. RFC 4346, Internet Engineering Task Force, April 2006. 25

[EIP] D. Erman, D. Ilie, and A. Popescu. Bittorrent session characteristics and models. *Traffic Engineering, Performance Evaluation Studies and Tools for Heterogeneous Networks*, 61(84):61. 14

[GDJ06a] S. Guha, N. Daswani, and R. Jain. An experimental study of the skype peer-to-peer voip system. In *In IPTPS 2006*, 2006. 72

[GDJ06b] Saikat Guha, Neil Daswani, and Ravi Jain. An Experimental Study of the Skype Peer-to-Peer VoIP System. In *Proceedings of The 5th International Workshop on Peer-to-Peer Systems (IPTPS '06)*, pages 1 – 6, Santa Barbara, CA, February 2006. 14

[GEBR+03] Luis Garces-Erice, Ernst W. Biersack, Keith W. Ross, Pascal A. Felber, and Guillaume Urvoy-Keller. Hierarchical p2p systems. In *Proceedings of*

ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par), 2003. xi, 18, 46

[GEFB+04] L. Garcs-Erice, P. A. Felber, E. W. Biersack, G. Urvoy-Keller, and K. W. Ross. Data indexing in peer-to-peer dht networks. *Distributed Computing Systems, International Conference on*, 0:200–208, 2004. 11

[GGGM04] Prasanna Ganesan, Krishna Gummadi, and Hector Garcia-Molina. Canon in g major: Designing dhts with hierarchical structure. *icdcs*, 00:263–272, 2004. 19, 20, 21, 46

[GM09] Y. Gao and Y. Meng. A new sip usage for reload, July 2009. Internet Draft draft-gaoyang-p2psip-new-sip-usage-00.txt. 25

[GMS04] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to-peer networks. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 1, pages –130, March 2004. 9

[gnu00] The gnutella protocol specification v0.4, 2000. `http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf`. 8, 9

[gnu02] Gnutella 0.6, 2002. `http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html`. 9, 13

[HRS+08] Mike Hibler, Robert Ricci, Leigh Stoller, Jonathon Duerig, Shashi Guruprasad, Tim Stack, Kirk Webb, and Jay Lepreau. Large-scale virtualization in the emulab network testbed. In Rebecca Isaacs and Yuanyuan Zhou, editors, *USENIX Annual Technical Conference*, pages 113–128. USENIX Association, 2008. 80

[Hua09] L. Huang. Location and discovery of subsets of resources, July 2009. Internet Draft draft-licanhuang-p2psip-subsetresourcelocation-01.txt. 40

[JEB09] X. Jiang, R. Even, and D. Bryan. An extension to reload to support direct response and relay peer routing, May 2009. Internet Draft draft-jiang-p2psip-relay-02.txt. 27

[JLR+09a] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne. Resource location and discovery (reload) base protocol, July 2009. Internet Draft draft-ietf-p2psip-base-03.txt. 23, 24, 25, 26, 27, 30, 39, 42, 44, 49, 79

[JLR+09b] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne. A sip usage for reload, March 2009. Internet Draft draft-ietf-p2psip-sip-01.txt. 25

[JPA04] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. RFC 3775, Internet Engineering Task Force, June 2004. 48

[KBPK05] Yoram Kulbak, Danny Bickson, Academic Prof, and Scott Kirkpatrick. The emule protocol specification. Technical report, DANSS (Distributed Algorithms, Networking and Secure Systems) Lab School of Computer Science and Engineering The Hebrew University of Jerusalem, Jerusalem, 2005. 8

[KF05] M. Kwon and S. Fahmy. Synergy: an overlay internetworking architecture. In *Computer Communications and Networks, 2005. ICCCN 2005. Proceedings. 14th International Conference on*, pages 401–406, 2005. 18

[KKO+09] Seok-Kap Ko, Young-Han Kim, Seung-Hun Oh, Byung-Tak Lee, and Victor Pascual Avila. Iptv usage for reload, July 2009. Internet Draft draft-softgear-p2psip-iptv-01.txt. 25

[LCC+02] Qin Lv, Pei Cao, Edith Cohen, Kai Li, and Scott Shenker. Search and replication in unstructured peer-to-peer networks. In *ICS '02: Proceedings of the 16th international conference on Supercomputing*, pages 84–95, New York, NY, USA, 2002. ACM. 9

[LCP+05] Eng Keong Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys & Tutorials, IEEE*, 7(2):72–93, 2005. 46

[Le09] Lifeng. Le. Hierarchical p2psip overlay, July 2009. Internet Draft draft-le-p2psip-hierachical-p2psip-overlay-00.txt. 40

[LHMZ05] Yin Li, Xinli Huang, Fanyuan Ma, and Futai Zou. *Grid and Cooperative Computing - GCC 2005 - LNCS*, chapter Building Efficient Super-Peer Overlay Network for DHT Systems, pages 787–798. Springer Berlin / Heidelberg, 2005. 10.1007/11590354_99. 21

[LSM+05] Jinyang Li, J. Stribling, R. Morris, M.F. Kaashoek, and T.M. Gil. A performance vs. cost framework for evaluating dht design tradeoffs under churn. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 1, pages 225–236 vol. 1, March 2005. 13, 50

[MB07] E. Marocco and D. Bryan. Interworking between p2psip overlays and conventional sip networks, March 2007. Internet Draft draft-marocco-p2psip-interwork-01.txt. 25

[MBRM06] A. MacQuire, A. Brampton, I.A. Rai, and L. Mathy. Performance analysis of stealth dht with mobile nodes. *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pages 5 pp.–189, March 2006. 49

[MCH09] J. Maenpaa, G. Camarillo, and J. Hautakorpi. A self-tuning distributed hash table (dht) for resource location and discovery (reload), February 2009. Internet Draft draft-maenpaa-p2psip-self-tuning-00.txt. 26

[MCKS03] A. T. Mizrak, Yuchung Cheng, Vineet Kumar, and S. Savage. Structured superpeers: leveraging heterogeneity to provide constant-time lookup. In *Internet Applications. WIAPP 2003. Proceedings.*, pages 104–111, 2003. 20, 43, 46, 72

[MHC06] Su-Hong Min, J. Holliday, and Dong-Sub Cho. Optimal super-peer selection for large-scale p2p system. In *Hybrid Information Technology, 2006. ICHIT'06. Vol 2. International Conference on*, volume 2, pages 588–593, 2006. 20, 21, 43, 46, 72

[MKL+02] Dejan S. Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins, and Zhichen Xu. Peer-to-peer computing. Technical report, HP, 2002. 7

[MM02] P. Maymounkov and D. Mazieres. *Peer-to-Peer Systems: First International Workshop, IPTPS 2002 Cambridge, MA, USA, March 7-8, 2002. Revised Papers*, volume 2429/2002 of *Lecture Notes in Computer Science*, chapter Kademlia: A peer-to-peer information system based on the XOR metric, pages 53–65. Springer, 2002. 10, 12, 19, 40, 65, 84, 89

[MS03a] J. Mischke and B. Stiller. Peer-to-peer overlay network management through agile. In *Integrated Network Management, 2003. IFIP/IEEE Eighth International Symposium on*, pages 337–350, 2003. 18

[MS03b] J. Mischke and B. Stiller. Rich and scalable peer-to-peer search with shark. In *Autonomic Computing Workshop, 2003*, pages 112–121, 2003. 18

[MSD+09] J. Maenpaa, A. Swaminathan, S. Das, G. Camarillo, and J. Hautakorpi. A topology plug-in for resource location and discovery, July 2009. Internet Draft draft-maenpaa-p2psip-topologyplugin-00.txt. 26

[MYBC+09] Isaias Martinez-Yelmo, Alex Bikfalvi, Ruben Cuevas, Carmen Guerrero, and Jaime Garcia. H-p2psip: Interconnection of p2psip domains for global multimedia services based on a hierarchical dht overlay network. *Computer Networks (Special Issue on Content Distribution Infrastructures for Community Networks)*, 53(4), Mar. 2009. 3, 39, 54, 55, 67, 71, 77, 82, 89, 95

[MYBG+08a] Isaias Martinez-Yelmo, Alex Bikfalvi, Carmen Guerrero, Ruben Cuevas, and Andreas Mauthe. Enabling global multimedia distributed services based on hierarchical dht overlay networks. *International Journal of Internet Protocol Technology (Special Issue on Future Multimedia Networking)*, 3(4), Dec. 2008. 3, 39, 54, 55, 67, 71, 77, 82, 89, 95

[MYBG+08b] Isaias Martinez-Yelmo, Alex Bikfalvi, Carmen Guerrero, Ruben Cuevas, and Andreas Mauthe. Enabling global multimedia distributed services based on hierarchical dht overlay networks. In *IEEE Conference on Next Generation Mobile Applications Services and Technologies 2008. Future Multimedia Networking Workshop*, pages 543–549. IEEE Computer Society, Sep. 2008. 3, 55, 67, 71, 77, 95

[MYBG09] Isaias Martinez-Yelmo, Alex Bikfalvi, and Carmen Guerrero. Benefits on using h-p2psip in mobile environments. In *JITEL 2009*, Sep. 2009. 47, 54

[MYCGM07] Isaias Martinez-Yelmo, Ruben Cuevas, Carmen Guerrero, and Andreas Mauthe. Analysis of searching mechanisms in hierarchical p2p based overlay networks. In *Med Hoc Net 2007*, Corfu, Greece, June 2007. 3, 95

[MYCGM08] Isaias Martinez-Yelmo, Ruben Cuevas, Carmen Guerrero, and Andreas Mauthe. Routing performance in hierarchical dht-based overlay networks. In *In Proceedings on 16th Euromicro International Conference on Parallel, Distributed and network-based Processing*, Feb. 2008. 3, 55, 67, 95

[MYGCM09] Isaias Martinez-Yelmo, Carmen Guerrero, Ruben Cuevas, and Andreas Mauthe. A hierarchical p2psip architecture to support skype-like services. In *Parallel, Distributed and Network-based Processing, 2009 17th Euromicro International Conference on*, pages 316–322, Feb. 2009. 3, 55, 67, 95

[Per02] C. Perkins. IP Mobility Support for IPv4. RFC 3220, Internet Engineering Task Force, January 2002. 48

[PGES05] Johan Pouwelse, Pawn Garbacki, Dick Epema, and Henk Sips. chapter The Bittorrent P2P File-Sharing System: Measurements and Analysis, pages 205–216. 2005. 10.1007/11558989_19. 14

[RD01] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218:329–351, 2001. 10, 12

[RFH+01] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, New York, NY, USA, 2001. ACM Press. 10, 13, 19, 40

[RGRK04] Sean Rhea, Dennis Geels, Timothy Roscoe, and John Kubiatowicz. Handling churn in a dht. In *ATEC '04: Proceedings of the annual conference on USENIX Annual Technical Conference*, pages 10–10, Berkeley, CA, USA, 2004. USENIX Association. 10, 15

[RM06] E. Rescorla and N. Modadugu. Datagram Transport Layer Security. RFC 4347, Internet Engineering Task Force, April 2006. 25

[RMM08] Dario Rossi, Marco Melia, and Michela Meo. A detailed measurment of skype network traffic. In *In IPTPS 2008*, 2008. 72

[Ros07] J. Rosenberg. Interactive connectivity establishment (ice): A protocol for network address translator (nat) traversal for offer/answer protocols, October 2007. Internet Draft draft-ietf-mmusic-ice-19.txt. 25

[RSC+02] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261, Internet Engineering Task Force, June 2002. 23

[SBEN07] Moritz Steiner, Ernst W Biersack, and Taoufik En Najjary. Actively monitoring peers in KAD. In *IPTPS'07, 6th International Workshop on Peer-to-Peer Systems, February 26-27, 2007, Bellevue, USA*, Feb 2007. 14, 72, 81

[SCMB05] H. Soliman, C. Castelluccia, K. El Malki, and L. Bellier. Hierarchical Mobile IPv6 Mobility Management (HMIPv6). RFC 4140, Internet Engineering Task Force, August 2005. 48

[SENB07a] Moritz Steiner, Taoufik En Najjary, and Ernst W Biersack. Analyzing peer behavior in KAD. Technical Report EURECOM+2358, Institut Eurecom, France, Oct 2007. 14, 72, 81

[SENB07b] Moritz Steiner, Taoufik En-Najjary, and Ernst W. Biersack. Exploiting kad: possible uses and misuses. *SIGCOMM Comput. Commun. Rev.*, 37(5):65–70, 2007. 14, 72

[SENB07c] Moritz Steiner, Taoufik En-Najjary, and Ernst W. Biersack. A global view of kad. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 117–122, New York, NY, USA, 2007. ACM. 14, 72, 81

[SJEB09] H. Song, X. Jiang, R. Even, and D. Bryan. P2psip overlay diagnostics, June 2009. Internet Draft draft-ietf-p2psip-diagnostics-01.txt. 25

[SM02] Emil Sit and Robert Morris. *Peer-to-Peer Systems*, chapter Security Considerations for Peer-to-Peer Distributed Hash Tables, pages 261–269. Springer Berlin / Heidelberg, 2002. 10.1007/3-540-45748-8_25. 9, 11

[SMLN+03] I. Stoica, R. Morris, D. Liben-Nowell, DR Karger, MF Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on networking*, 11(1):17–32, 2003. 10, 11, 26, 40

[SR06] Daniel Stutzbach and Reza Rejaie. Understanding churn in peer-to-peer networks. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 189–202, New York, NY, USA, 2006. ACM. 14

[UPvS09] Guido Urdaneta, Guillaume Pierre, and Maarten van Steen. A survey of DHT security techniques. *ACM Computing Surveys*, 2009. `http://www.globule.org/publi/SDST_acmcs2009.html`, to appear. 42

[VYW+02] Amin Vahdat, Ken Yocum, Kevin Walsh, Priya Mahadevan, Dejan Kostić, Jeff Chase, and David Becker. Scalability and accuracy in a large-scale network emulator. *SIGOPS Oper. Syst. Rev.*, 36(SI):271–284, 2002. 80

[Wal03] Dan Wallach. *Software Security. Theories and Systems*, chapter A Survey of Peer-to-Peer Security Issues, pages 253–258. Springer Berlin / Heidelberg, 2003. 10.1007/3-540-36532-X_4. 9, 11

[WSM09] J. Wang, J. Shen, and Y. Meng. Content sharing usage for reload, June 2009. Internet Draft draft-shen-p2psip-content-sharing-00.txt. 25

[XMH03] Zhiyong Xu, Rui Min, and Yiming Hu. Hieras: a dht based hierarchical p2p routing algorithm. In *Parallel Processing, 2003. Proceedings. 2003 International Conference on*, 2003. 19, 20, 21, 46

[XZ02] Zhichen Xu and Zheng Zhang. Building low-maintenance expressways for p2p systems. Technical report, Internet Systems and Storage Laboratory. HP Laboratories Palo Alto, 2002. 19, 20, 21

[ZDK06] S. Zoels, Z. Despotovic, and W. Kellerer. Cost-based analysis of hierarchical dht design. In *Peer-to-Peer Computing, 2006. P2P 2006. Sixth IEEE International Conference on*, pages 233–239, 2006. 20, 46

[ZHS+04] B.Y. Zhao, L. Huang, J. Stribling, S.C. Rhea, A.D. Joseph, and J.D. Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *IEEE Journal on selected areas in communications*, 22(1):41–53, 2004. 10

[ZWH03] Y. Zhu, H. Wang, and Y. Hu. A super-peer based lookup in structured peer-to-peer systems. In *Proceedings of the 16th International Conference on Parallel and Distributed Computing Systems (PDCS'03)*. Citeseer, 2003. 21

[ZWL07] Xiao-Ming Zhang, Yi-Jie Wang, and ZhouJun Li. *LNCS: Parallel and Distributed Processing and Applications*, chapter Research of Routing Algorithm in Hierarchy-Adaptive P2P Systems, pages 728–739. Springer Berlin / Heidelberg, 2007. 19, 20, 21

Acronyms

CoA	Care-of Address
DHT	Distributed Hash Table
DTLS	Datagram Transport Layer Security protocol
HA	Home Agent
HoA	Home Address
IANA	Internet Assigned Numbers Authority
ID	Identifier
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPTV	Internet Protocol Television
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
NAT	Network Address Translation
p2p	Peer-to-Peer
RELOAD	REsource LOcation And Discovery base protocol
SIP	Session Initiation Protocol
SP	Service Provider
STUN	Simple Transversal of UDP over NAT's
TLS	Transport Layer Security protocol
TTL	Time to Live

TURN Traversal Using Relay NAT

URI Uniform Resource Identifier

VoD Voice on Demand

WG Working Group